ASTES

Advances in Science, Technology & Engineering Systems Journal

VOLUME 9-ISSUE 5|SEPT-OCT 2024

www.astesj.com
ISSN: 2415-6698

Editorial

The continued advancement of digital technologies, artificial intelligence, and education-support systems has paved the way for innovative research that bridges theoretical insight and practical implementation. This issue presents a collection of compelling studies that highlight novel approaches to challenges in software engineering, AI transparency, classroom education, and hardware security. The diversity of contributions reflects a shared commitment to usability, ethical compliance, and technological resilience—values that are increasingly important in today's interconnected and rapidly evolving research environment.

Addressing the chronic problem of inefficient and incomplete bug reporting in self-hosted systems, a lightweight framework named *Watson* is introduced to enhance developer workflows. By capturing user interactions, screen recordings, and network activity, Watson minimizes the user's effort during the reporting process while significantly increasing the quality of reports. Its seamless integration with issue trackers, without relying on cloud services or external APIs, makes it ideal for confidentiality-sensitive environments. Experimental evaluations show that Watson triples the efficiency in identifying root causes of bugs compared to traditional manual reporting, suggesting a promising direction for modernizing software maintenance tools [1].

Explainable Artificial Intelligence (XAI) remains a cornerstone in the development of transparent and accountable AI systems. A comprehensive exploration of XAI concepts reveals the nuanced difference between explainability and interpretability, while shedding light on cutting-edge techniques like feature attribution and rule extraction from neural networks. The discussion extends to the regulatory landscape, emphasizing the urgent need for governance structures that can evolve in tandem with rapid AI developments. The work not only advances academic discourse on AI ethics but also proposes pragmatic considerations for policy and research in high-stakes domains such as finance and healthcare [2].

Innovations in programming education are also represented in this volume, with the development of a classroom support system that complements tangible educational tools. Designed for real-time monitoring of student progress, the system helps instructors identify common learning barriers and deliver timely, tailored guidance. Deployed in a high school setting, the tool was well-received by both teachers and students, who appreciated its ability to personalize the learning experience. While limitations remain in the rigidity of predefined model answers, this system marks a significant step toward scalable and data-informed teaching practices in computer science education [3].

In the domain of hardware-based security, a novel approach to true random number generation is presented using Resistive Switching Random Access Memories (ReRAMs). By comparing the high-resistance states of two ReRAM devices, the design avoids the precision timing constraints found in other TRNGs. Fully compatible with existing ReRAM crossbar architectures, the generator passed the NIST randomness test suite, validating its performance. Moreover, the analysis of device-to-device variability offers insight into the robustness of this approach, paving the way for secure and efficient random number generation in hardware cryptographic systems [4].

Collectively, these studies illustrate the breadth and depth of modern research efforts aimed at improving system reliability, user-centered design, educational innovation, and digital trust. By tackling domain-specific problems with interdisciplinary solutions, these contributions move us closer to a more intelligent, secure, and equitable technological future.

References:

[1] D. Costa, G. Matos, A. Lins, L. Barroso, C. Aguiar, E. Bezerra, “Web Application Interface Data Collector for Issue Reporting,” Advances in Science, Technology and Engineering Systems Journal, 9(5), 1–8, 2024, doi:10.25046/aj090501.

[2] M. Leon, H. DeSimone, “Advancements in Explainable Artificial Intelligence for Enhanced Trans- parency and Interpretability across Business Applications,” Advances in Science, Technology and Engineering Systems Journal, 9(5), 9–20, 2024, doi:10.25046/aj090502.

[3] K. Oda, T. Kato, Y. Kambayashi, “Evaluation of a Classroom Support System for Programming Education Using Tangible Materials,” Advances in Science, Technology and Engineering Systems Journal, 9(5), 21–29, 2024, doi:10.25046/aj090503.

[4] T. Patni, A. Pethe, “True Random Number Generator Implemented in ReRAM Crossbar Based on Static Stochasticity of ReRAMs,” Advances in Science, Technology and Engineering Systems Journal, 9(5), 30–36, 2024, doi:10.25046/aj090504.

Editor-in-chief

Prof. Passerini Kazmersk

ADVANCES IN SCIENCE, TECHNOLOGY AND ENGINEERING SYSTEMS JOURNAL

Volume 9 Issue 5 September-October 2024

CONTENTS

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 9, No. 5, 1-8 (2024)
www.astesj.com
Special Issue on Computing, Engineering and Multidisciplinary Sciences

ASTES Journal
ISSN: 2415-6698

Web Application Interface Data Collector for Issue Reporting

Diego Costa*, Gabriel Matos, Anderson Lins, Leon Barroso, Carlos Aguiar, Erick Bezerra

SIDIA Institute of Science and Technology, Manaus, Brazil

ARTICLE INFO

Article history:
Received: 24 April, 2024
Revised: 01 August, 2024
Accepted: 03 August, 2024
Online: 14 September, 2024

Keywords:
Bug Reporting
Software Management
Web Application
Browser API

ABSTRACT

Insufficient information is often pointed out as one of the main problems with bug reports as most bugs are reported manually, they lack detailed information describing steps to reproduce the unexpected behavior, leading to increased time and effort for developers to reproduce and fix bugs. Current bug reporting systems lack support for self-hosted systems that cannot access third-party cloud environments or Application Programming Interfaces due to confidentiality concerns. To address this, we propose Watson, a Typescript framework with a minimalist User Interface developed in Vue.js. The objectives are to minimize the user's effort to report bugs, simplify the bug reporting process, and provide relevant information for developers to solve it. Watson was designed to capture user's interactions, network logs, screen recording, and seamlessly integration with issue tracker systems in self-hosted systems that cannot share their data to external Application Programming Interfaces or cloud services. Watson also can be installed via Node Package Manager and integrated into most JavaScript or TypeScript web projects. To evaluate Watson, we developed an Angular-based application along with two usage scenarios. First, the users experimented the application without using Watson and once they found a bug, they reported it manually on GitLab. Later, they used the same application, but this time, whenever they detect another bug, they reported it through Watson User Interface. Watson, as stated by the experiment participants and the evidences, is useful and helpful for development teams to report issues and provide relevant information for tracking bugs. The identification of bug root causes was almost three times more effective with Watson than manual reporting.

1. Introduction

This paper is an extension of work originally presented at the 2023 IEEE 30th Annual Software Technology Conference (STC) [1]. Bug reports play a crucial role in software maintenance, enabling developers to prioritize, reproduce, identify, and resolve defects [2], [3]. Detailed information is expected from the reports, such as the unexpected behavior, the steps to reproduce it, logs, or screenshots, and others, so developers may recreate it to find a solution [2, 4, 5].

Insufficient information is often pointed out as one of the main problems with bug reports, generally most bugs are reported manually by end-users or testers, the reports lack of details and sufficient information describing the steps to reproduce the unexpected behavior to allow the developers to find a solution [2, 6].

The most common way to report bugs is through issue tracking systems, but in the majority cases, there is not any standard, which causes misinformation for the development team due the unclear or insufficient data [7]. Steps to reproduce the bug, stack trace errors, test case scenarios, logs, and images are factors that impact the quality of the bug reports [6, 8, 9, 10, 11, 12].

In this paper, we introduce Watson, a framework developed in Typescript[1] with an User Interface (UI) developed in Vue.js[2]. The objective is to save time and effort for the person who is going to report the bug, and standardize bug reporting by collecting fundamental information that will aid the developers to reproduce the undesired behavior, as such as: user interaction on the page, network requests, and screen video. The main points of Watson are that it is a framework that can be installed via Node Package Manager (NPM), it can be integrated into most Javascript[3] or Typescript web projects, and it is designed for self-hosted systems that cannot share its data to external Application Programming Interfaces (APIs) or cloud services.

*Corresponding Author: Diego Costa, SIDIA Institute of Science and Technology, Manaus, Brazil, diego.costa@sidia.com
[1]https://www.typescriptlang.org/
[2]https://vuejs.org/
[3]https://developer.mozilla.org/en-US/docs/Web/JavaScript

Watson was evaluated empirically, using a test application developed in Angular[4] and inviting developers and testers to use it with Watson. The goal was to evaluate the participant's perception of Watson usefulness in comparison with manually reporting a bug on GitLab[5]. Experienced web developers an testers judged Watson as useful and Watson's features to collect information proved helpful in identifying the root cause of bugs reported.

The rest of this work is organized as follows: Section 2 provides details about the problem and related work. Section 3 provides details about the proposed software. Section 4 presents the results of the use of Watson. Section 5 concludes by discussing the main points found.

2. Related Works

Our research focused on bug reporting tools that provided relevant and effective information for software maintenance, mainly for web applications, but due to the lack of recent works, we expanded the search for Android applications and approaches that are emerging, such as machine learning.

The authors of [13] propose an automated bug reporting system which serves as a foundation for testing frameworks in web applications, generating failure reports based on the test cases. The generated reports consist of the number of test cases executed, the number of failed, passed and skipped tests, and the time it took to perform the tests. When the report is ready, it is checked if it is a duplicated bug report, then it is mailed to the development team. The tool was used by the authors to simulate regression test cases, resulting in an 8% reduction in test execution time. Additionally, it summarized the bug report, reducing human effort and time spent filtering duplicated bug reports.

In the work [14], they created a tool, Euler, that automatically analyzes the written description of a bug report, evaluates the quality of reproduction steps, and provides feedback to users about ambiguous or missing steps. Neural sequence labeling combined with discourse patterns and dependency parsing identifies sentences and individual steps to reproduce. It matches these steps to program state and Graphical User Interface (GUI) in a graph-based execution model. An empirical evaluation was conducted to determine the accuracy of Euler in identifying and assessing the quality of reproduction steps for bug reports. The results indicated that Euler correctly identified 98% of the existing steps to reproduce and 58% of the missing ones, and 73% of its quality annotations being accurate.

The work [2] presents Bee, an open-source tool that can be integrated with GitHub[6] and automatically analyzes user-written bug reports using machine learning textual classification. It offers insights into the system's observed comportment, expected comportment, and reproduction steps for the reported bugs. As result they achieved 87% recall, indicating the ability to correctly detect and classify the described sentences.

In order to generate better bug reports for Android, CrashScope [15] was created. It works by collecting system version and hardware information, the application state, the user entry text description to reproduce the bug, the GUI events, and app's stack trace error. The purpose was to assess the reliability and comprehensibility of reports generated by CrashScope relative to five current tools. To evaluate this work, they used 8 real world open source applications bug reports extracted from their corresponding issue trackers. They invited 16 users to reproduce 4 bugs reported using CrashScope and 4 bugs reported manually. It was discovered that reports produced by CrashScope were equally reproducible compared to those from other tools, although it yielded more comprehensible and beneficial reports for developers.

In [16], the author presented a chatbot that designed for Android, combines dynamic software analysis, natural language processing, and automated report quality assessment to assist users in writing better descriptions and receiving issue reports. By inviting 18 end-users to identify 12 bugs across 6 Android apps, we found that Burt provides more accurate and complete reproduction steps than Itrac, a template-based bug reporting system used by another 18 users.

The work [17] proposes an automated tool for integrating user feedback into the testing process. In order to achieve this, they collected datasets of mobile application issues reviews along with user feedbacks to train a machine learning algorithm that would be capable to link the user's feedback to stack traces with the objective to relate a feedback that might describe the cause of a failure to a bug. By following this process, they concluded user feedback is highly promising to integrate into the testing process of mobile apps as it complements the capabilities of testing tools identifying bugs that are not reveled in this phase of software development, facilitating the diagnosis of bugs or crashes.

In the study [18], the authors explored automating Android bug replaying using Large Language Models (LLMs). Motivated by the success of these models, they proposed AdbGPT, a method for reproducing bugs from bug reports via prompt engineering. By following this approach, they demonstrated 81.3% effectiveness and efficiency to reproduce bugs from users' reports, and in terms of average time to reproduce bug reports, the AdbGPT outperformed the average time of experiment participants.

Other studies on bug reporting tools for web applications [19, 20] and commercial tools like Usersnap[7], BugHerd[8], and Bird Eats Bugs[9] utilize browser extensions or embedded scripts to capture Document Object Model (DOM) events, stack trace errors, and screenshots. Data is sent to cloud services but lacks integration with self-hosted systems that restrict sending its private data to external APIs. This issue is addressed by Watson, a framework installable via NPM for JavaScript or TypeScript web projects. It provides flexibility for development teams to integrate and utilize self-hosted systems, enabling configuration of Watson to use entirely their own systems while collecting crash report data without external environments.

[4] https://angularjs.org/
[5] https://about.gitlab.com/
[6] https://github.com
[7] https://usersnap.com/
[8] https://bugherd.com/
[9] https://birdeatsbug.com/

3. Watson Framework

Watson was developed in Typescript, offering an API to help collect important information. Watson collects information such as DOM events related to the user's interaction, when a user interacts with the application, Watson interceptors will collect the web page events, such as mouse clicks, network requests and screen video to capture what the user sees on the page. This information will be passed to the reporter class that implements the interface WatsonReporter to attach the information to a bug report with a description provided by the user and the Watson collected data, then it will send to the chosen issue tracking system.

As presented in Figure 1, Watson acts in the native browser API, injecting a collection of interceptors that will intercept data during web application usage. When the user initiates the recording process through the UI, Watson begins collecting data. Upon completion of the recording, the gathered information can be transmitted to an issue tracking system based on the preconfigured reporter implementation. This may involve utilizing a built-in reporter such as GitLabReporter or implementing a customized reporter to facilitate integration with alternative issue trackers or systems.

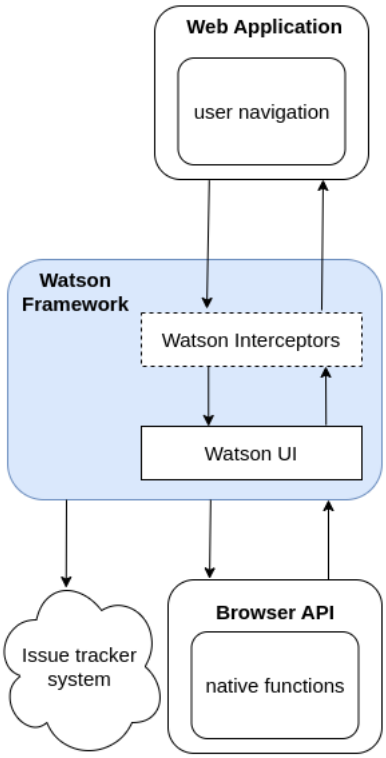
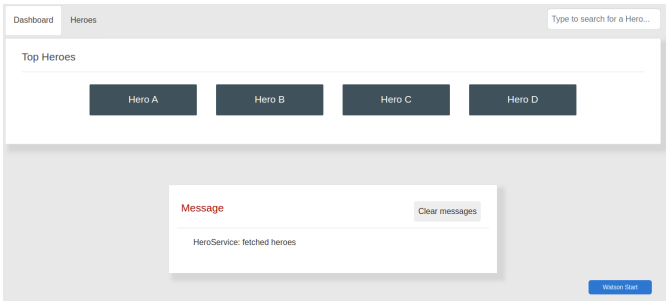
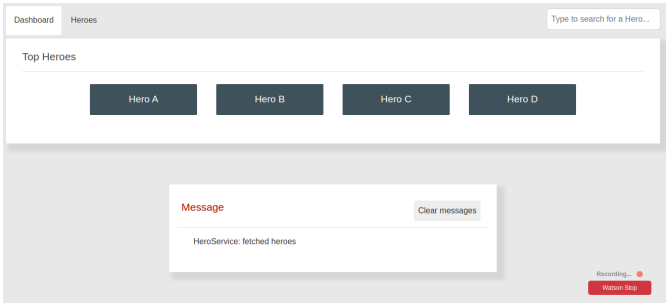


Figure 1: Watson Architecture

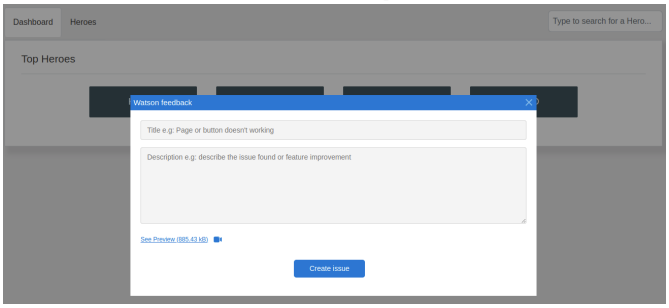
The UI workflow used to report bugs with Watson is shown in Figure 2. Starting with the Watson UI start button (Figure 2a), users can proceed as normal in their testing scenario or regular application usage. Once Watson is running, it attaches interceptors to the browser's native API, listens for native events, collects data, and redirects event parameters to the original event calls as proxies, allowing it to intercept and save the user interaction with the web page.



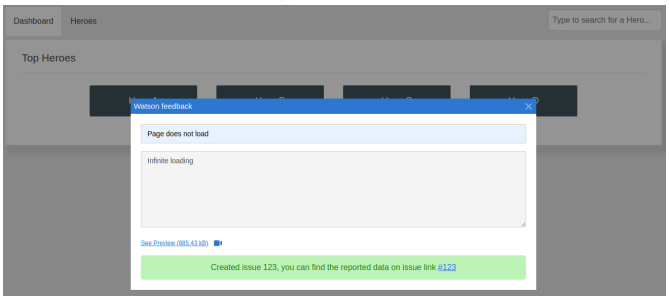
(a) Watson start



(b) Watson stop



(c) Create issue



(d) Issue created

Figure 2: Watson UI

After the data is intercepted, it gets transferred to the initial web browser function API to perform its regular actions. To stop the data collection, the user can use the stop button on Watson UI (Figure 2b), then, as shown in Figure 2c, a dialog message opens for users to provide additional description about the bug and a title for the issue, which will be sent to an external issue tracking system such as GitHub[10], Jira[11] or another one which the developers may have

[10]https://github.com
[11]https://www.atlassian.com/software/jira

previously integrated Watson to it. As in this case, it was integrated with GitLab, it returned the issue link in Figure 2d.

3.1. Implementation details

Figure 3 presents Watson's class diagram. It is composed of four components: **Watson**, **WatsonConfig**, **WatsonReporter**, and **Watson interceptors**.

Watson can be installed on a web project with NPM. After installation, the Watson instance is a singleton and must be initialized at application start-up, initializing and attaching its injectors to the browser's native API, functioning as proxies. The Watson framework must be initialized and configured by adding the code **Watson.getInstance(config)** on application startup code. The config paramter is object of type **WatsonConfig** that contains which interceptors should be active and the reporter implementation of **WatsonReporter** interface that has the responsibility to connect Watson to an external issue tracking system. This enables reporting of collected information alongside user description and issue title.

3.1.1. Watson

The Watson class connects all parts of the system and ensures the data capture functions properly. It manages the start and stop of data capture, stores the collected data, controls the event interceptors activating and deactivating the capture and also uses the reporter instance to send the collected data to issue tracker. With the config parameter, it enables which event interceptors that will be attached to native API and the event types that should be captured, and also defines the WatsonReporter instance to send the collected data.

3.1.2. WatsonConfig

The configuration interface determines which event interceptors should be activated. Currently, there are 3 event interceptors to capture user interaction data: network interceptor, DOM events interceptor, and the screen recorder. By default, all interceptors are enabled, but the development team can define which of them are active according to their requirements. For example, to disable screen recorder by setting **screenRecorder** option to false. Additionally, the implementation instance of the WatsonReporter interface must be specified in the config. This can utilize either a built-in implementation such as GitlabReporter or a custom report class that implements the interface.

3.1.3. WatsonReporter

This is the interface implementation required for reporting to external issue tracking system. A method must be implemented to send the collected data to an external issue tracking system like Jira, GitLab, or others. The development team can write its code to connect to the external system and implement this interface to send data. For this experiment, Figure 3 presents the GitLabReporter class designed to send the collected data to GitLab.

To implement this interface, the development team must first obtain access to the required issue tracking system API and follow its documentation to consume its API and have information about it such as API key, authentication, and available endpoints. Following this step, the developers can implement a class with this interface containing the methods to communicate with the issue tracking system and receive the data collected by Watson to finally transmit them.

The WatsonReporter interface has a method called **reportData**, accepting two parameters. The first parameter consists of the issue's basic information, including its title and description, while the second one is the Watson collected data. This function is responsible for generating issues within the issue tracker and returning an object containing the issue's ID and corresponding link, or null in case of failure.

3.1.4. Watson Interceptors

Watson interceptors are built-in functions that will be attached on native API, in order to listen network requests and DOM events

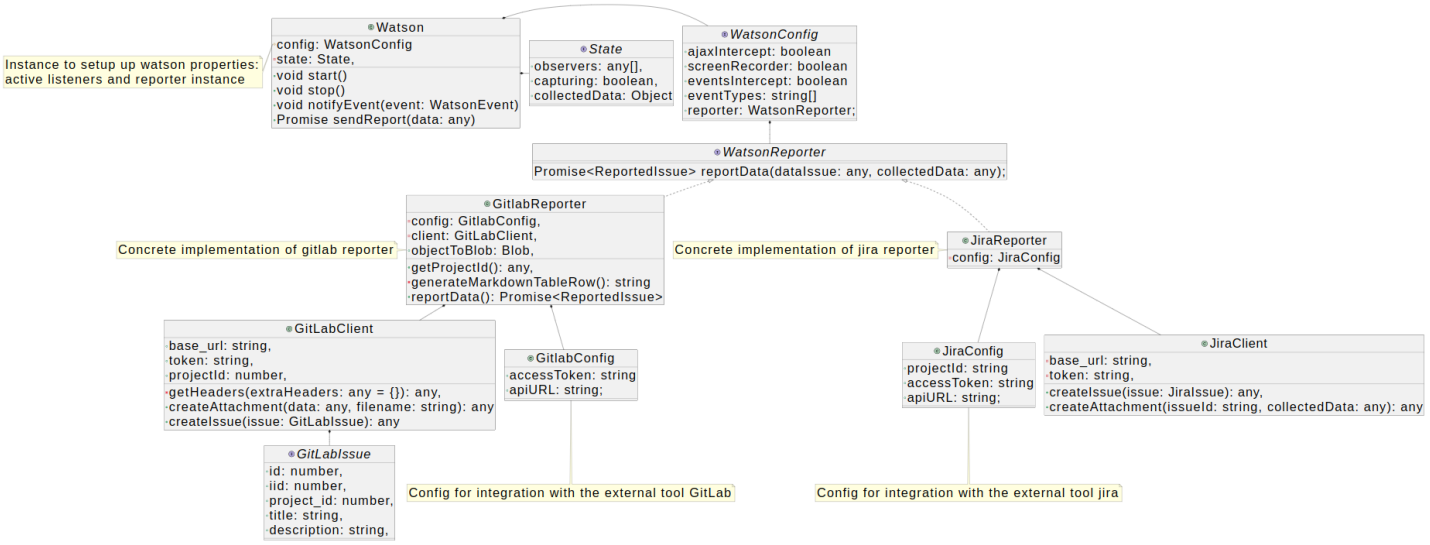


Figure 3: Watson Class Diagram

specified in the event types list, according to active event interceptors specified in the Watsonconfig when Watson is instantiated. These functions will be added to a list of observables at Watson states. Watson has three types of interceptors:

- **Video recording**: Watson utilizes the MediaStream API[12] to capture the user's screen activity. By clicking the start button within the Watson UI, the user initiates the recording process; while stopping it requires clicking the stop button. Besides bug reporting, this can be used to record test scenarios. The recorded video will be attached in the issue report to be sent along with the logs.
- **Network requests**: As represented in Figure 4, Watson injects the ajax interceptor to the XMLHttpRequests API[13] for collecting vital data about network requests. As soon the application is in use, it sends Hypertext Transfer Protocol (HTTP) requests based on user interactions. The native API is proxied, allowing the interceptor to gather information about the network requests such as headers, start and end time, status codes, sent and received data, and request duration. All collected information is stored and used to generate a JSON log file, which will be attached on the issue report.

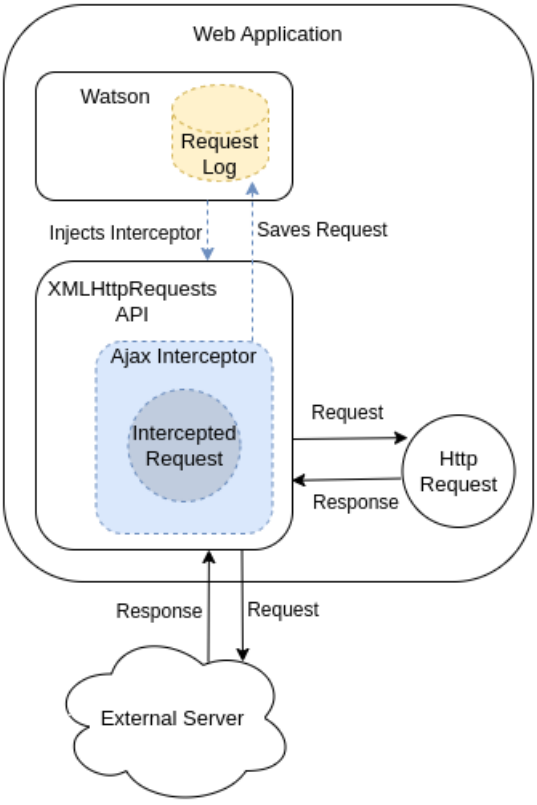


Figure 4: Watson - Network Interceptor

- **Capturing DOM events**: In order to collect the user's interaction with the web application, such as mouse and keyboard actions, Watson injects its interceptor into the native API[14] of AddEventListener, gathering data on events such as clicks, double clicks, mouse enter, mouse leave along with element details including HyperText Markup Language (HTML) tag name, node xpath, text content, as shown be seen in Figure 5.

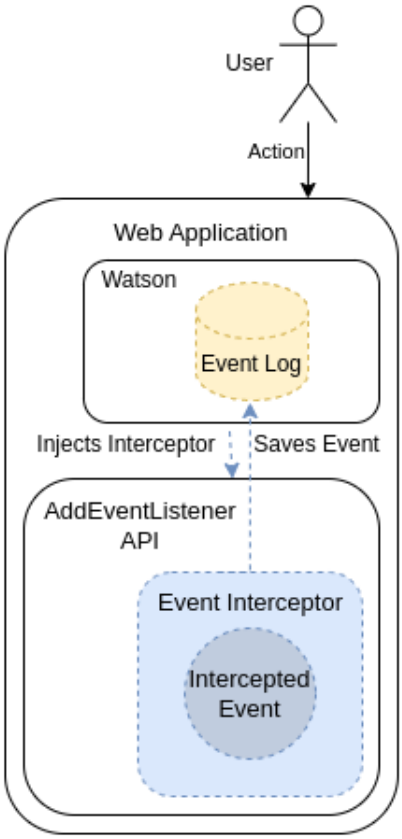


Figure 5: Watson - Event Interceptor

3.1.5. *GitlabReporter*

It is a concrete implementation of interface WatsonReporter, it can create an issue on Gitlab through its Web API and upload the collected information to the created issue.

The components of the reporting system: GitLabClient, GitLabConfig, and GitLabIssue. Each component is detailed below:

GitlabConfig serves as the interface to represent the minimum information necessary to identify the project and access it through the Gitlab API. It requires the server API url, as it may be a self-hosted server, and a valid access token registered to a Gitlab bot authorized to create issues within the project. These information are necessary during the configuring of the GitlabReporter as a WatsonReporter instance at application launch, all those configuration must be provided and setting up by the development team.

GitlabClient is responsible to use the Gitlab Web API use the GitlabConfig information. It is able to create issues and upload attachments to an issue.

[12]https://developer.mozilla.org/en-US/docs/Web/API/MediaStream
[13]https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest
[14]https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener

GitlabIssue Is a representation of represents a created issue on Gitlab. It contains basic issue details like issue ID, project ID, title and description.

4. Test Case

A web application was developed with Angular to evaluate the Watson framework. Watson was set up through NPM, configured and incorporated into the application accordingly. As previously mentioned, a GitLab reporter class that implemented the interface for Watson reporter was created to integrate and report the issue to Gitlab.

As in the works of [3, 14, 15, 16, 21], an empirical experiment was conducted in two parts to evaluate if the users enjoyed Watson's functionalities and if these would be relevant for debugging and solving issues.

As described in Table 1, an online questionnaire was created for the users to answer after completing the tests. This questionnaire aimed to assess user perceptions of Watson's features and effectiveness in resolving bugs relative to manual reporting. Quantitative and qualitative questions were included, with participants able to respond on a scale of 0 to 5, providing feedback on areas for improvement and suggestions.

Table 1: Questionnaire

1	Considering the manual method of creating an issue, how error-prone do you think this method is?
2	Considering the Watson method of creating an issue, how error-prone do you think this method is?
3	What would be the main reasons to use the manual method?
4	What would be the main reasons to use Watson?
5	Considering the method using Watson, how much would the network log be relevant to debug?
6	Considering the method using Watson, how much would DOM events be relevant to debug?
7	Considering the method using Watson, how much would the video be relevant to debug?
8	How relevant would Watson be to your project's issue report?
9	If you use or know other issue trackers. How much value would Watson add as a library compared to other issue trackers?
10	Do you have any suggestions for improvement? If yes, which one?

Six web developers and six testers were invited to try the application, discover, and report bugs. Some bugs were found in the web application, including issues with the network connection to database, visual elements, and behavior. The participants were not told about which bugs were incorporated into the application, and Watson was installed and configured by us.

In the fist part of the experiment, the users tried the web application without Watson installed, and once they noticed a bug, they manually reported it on GitLab, describing and attaching anything they judged necessary to solve the bug. In the second part, they tried the application with Watson installed. As in the first part, they explored the system until they found a bug, but this time, with Watson available in the application, they used Watson's UI to report the bug.

Upon completion of testing the application across both scenarios, participants responded to an online survey assessing their satisfaction with Watson's functionalities and the relevance of data gathered by Watson towards addressing reported bugs.

The purpose of the first and second questions was to evaluate how error-prone manual report and the Watson report were. For these two questions, the lower value is better because it indicates less error-prone than a higher value. For Watson method, the average grade was 2.22, and the manual method was 3.11.

The third and fourth questions were about the motives to use one method over another. The distribution about the main reasons the participants considered using the manual method was three votes for duplicate issue report prevention, two votes for organization, one vote for quickly generating evidences, one vote for using a text editor, and one vote for less-error prone, indicating that the participants noticed the prevention of duplicate issues more. For Watson, the results were of 5 votes for taking less time to report an issue, two votes for more bug evidences and two votes for less-error prone, indicating that the participant's preferred Watson because they took less time to report an issue.

The fifth, sixth, and seventh questions are about the relevance of Watson features: network logs, DOM events, and screen records collected by Watson. The average rates for these features were 4.11, 5 and 4.78, respectively.

The eighth question asks about the relevance of Watson for the participant's project issue report. The average rate for it was 4.89. The ninth question asks to compare Watson's reports to other tools if the participant used another one. The average rate for it was 4.22.

The tenth question asks for suggestions. The suggestions the participants cited were to add a text editor for the Watson's description field, an option to download the logs and generated video, an option to choose which project they would like to report the issue and label it, and mainly the history search to avoid duplicate issues before sending the new issue.

To evaluate the efficiency of Watson compared to manual reporting, all the bug reports were evaluated and analyzed for its potential to identify the root cause of the issue. This evaluation involved utilizing videos and user descriptions to recreate the bug scenario, as well as examining the collected stack traces to trace the faulty functionality.

Based on evidence provided by participants through manual bug reporting, it was found the bug root cause on 18.75% of the manually reported issues. In contrast, using Watson's reports containing stack traces and video, the number of issues that could be determined the bug root cause increased to 63.63%.

The recorded video helped to reproduce all of the bug reported with Watson. Additionally, the event and network logs were used to investigate the application's behavior during the occurrence of these bugs.

5. Conclusion

This work presented Watson, a software program developed to standardize the bug report process. Its main purposes are to reduce the

effort to collect bug evidences and provide relevant information for developers to solve it.

Experienced web developers and testers tried Watson and perceived it as useful and relevant for crash reports. Watson scored an average rate of 4.89 out of 5 from the participants questionnaire when asked about its impact on issue reporting in their web projects.

Watson's features for collecting event and network stack traces proved useful in identifying the root cause of bugs in 63.63% of cases when using Watson, against to only 18.75% of the issues reported manually.

This study focused on Watson's ability to gather necessary data for bug reports and make easier for user to report bugs with evidences. Unlike from many other reporting tools, Watson operates independently of any browser plugins and does not require sending data to external servers.

The results of the analysis provided by this article are limited and defined by the test case performed. To interpret the knowledge obtained as a general approach, additional use cases in different projects are needed. Therefore, our proposal suits the need to use a bug reporting tool without using a third-party cloud, which is a requirement for scenarios where highly confidential projects are developed.

As points of improvement mentioned by the participants, most of them were related to the user experience, such as the text editor, options to download collected information, and the recorded screen. It was also suggested to Watson block duplicated bugs.

The main achievement of this project was creating a framework using TypeScript, which can be easily incorporated into web applications and various issue tracking systems. This enables development teams to conveniently obtain standardized bug reports and user feedback. Projects can benefit from this solution as it allows them to bypass third-party servers and incorporate a reporting tool within their self-hosted system, especially when handling sensitive data.

As future work, besides the already mentioned user experience suggestions, we plan to develop a mechanism to prevent duplicate bug reports using Watson and machine learning techniques.

Acknowledgment This work is the result of the R&D project *Projeto de Engenharia de Software e Ciência de Dados aplicados ao Desenvolvimento de Sistemas*, performed by Sidia Instituto de Ciência e Tecnologia in partnership with Samsung Eletrônica da Amazônia Ltda., using resources from Federal Law No. 8.387/1991, and its disclosure and publicity are under the provisions of Article 39 of Decree No. 10.521/2020. Rodrigo José Borba Fernandes, Isabelle Maria Farias de Lima Teixeira, and Thiago Cruz Ferraz former members.

References

[1] G. Matos, D. Costa, A. Lins, E. Bezerra, L. Barroso, C. Aguiar, T. Ferraz, I. Teixeira, "Watson: Web Application Interface Data Collector for Feedback Reporting," in 2023 IEEE 30th Annual Software Technology Conference (STC), 3–6, 2023, doi:10.1109/STC58598.2023.00007.

[2] Y. Song, O. Chaparro, "Bee: A tool for structuring and analyzing bug reports," in Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 1551–1555, 2020, doi:10.1145/3368089.3417928.

[3] K. Moran, "Enhancing android application bug reporting," in Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, 1045–1047, 2015, doi:10.1145/2786805.2807557.

[4] M. Erfani Joorabchi, M. Mirzaaghaei, A. Mesbah, "Works for me! characterizing non-reproducible bug reports," in Proceedings of the 11th working conference on mining software repositories, 62–71, 2014, doi:10.1145/2597073.2597098.

[5] M. Soltani, F. Hermans, T. Bäck, "The significance of bug report elements," Empirical Software Engineering, **25**, 5255–5294, 2020, doi:10.1007/s10664-020-09882-z.

[6] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schroter, C. Weiss, "What makes a good bug report?" IEEE Transactions on Software Engineering, **36**(5), 618–643, 2010, doi:10.1109/TSE.2010.63.

[7] D. Huo, T. Ding, C. McMillan, M. Gethers, "An empirical study of the effects of expert knowledge on bug reports," in 2014 IEEE International Conference on Software Maintenance and Evolution, 1–10, IEEE, 2014, doi:10.1109/ICSME.2014.22.

[8] J. Wang, M. Li, S. Wang, T. Menzies, Q. Wang, "Images don't lie: Duplicate crowdtesting reports detection with screenshot information," Information and Software Technology, **110**, 139–155, 2019, doi:10.1016/j.infsof.2019.03.003.

[9] N. Cooper, C. Bernal-Cárdenas, O. Chaparro, K. Moran, D. Poshyvanyk, "It Takes Two to Tango: Combining Visual and Textual Information for Detecting Duplicate Video-Based Bug Reports," in 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 957–969, 2021, doi:10.1109/ICSE43902.2021.00091.

[10] M. Bheree, J. Anvik, "Identifying and Detecting Inaccurate Stack Traces in Bug Reports," in 2024 7th International Conference on Software and System Engineering (ICoSSE), 9–14, IEEE Computer Society, Los Alamitos, CA, USA, 2024, doi:10.1109/ICoSSE62619.2024.00010.

[11] Y. Noyori, H. Washizaki, Y. Fukazawa, K. Ooshima, H. Kanuka, S. Nojiri, "Deep learning and gradient-based extraction of bug report features related to bug fixing time," Frontiers in Computer Science, **5**, 1032440, 2023, doi:10.3389/fcomp.2023.1032440.

[12] R. Krasniqi, H. Do, "A multi-model framework for semantically enhancing detection of quality-related bug report descriptions," Empirical Software Engineering, **28**(2), 42, 2023, doi:10.1007/s10664-022-10280-w.

[13] Y. Sharma, A. Dagur, R. Chaturvedi, et al., "Automated bug reporting system in web applications," in 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 1484–1488, IEEE, 2018, doi:10.1109/ICOEI.2018.8553850.

[14] O. Chaparro, C. Bernal-Cárdenas, J. Lu, K. Moran, A. Marcus, M. Di Penta, D. Poshyvanyk, V. Ng, "Assessing the quality of the steps to reproduce in bug reports," in Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 86–96, 2019, doi:10.1145/3338906.3338947.

[15] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, C. Vendome, D. Poshyvanyk, "Automatically discovering, reporting and reproducing android application crashes," in 2016 IEEE international conference on software testing, verification and validation (icst), 33–44, IEEE, 2016, doi:10.1109/ICST.2016.34.

[16] Y. Song, J. Mahmud, Y. Zhou, O. Chaparro, K. Moran, A. Marcus, D. Poshyvanyk, "Toward interactive bug reporting for (android app) end-users," in Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 344–356, 2022, doi:10.1145/3540250.3549131.

[17] G. Grano, A. Ciurumelea, S. Panichella, F. Palomba, H. C. Gall, "Exploring the integration of user feedback in automated testing of Android applications," in 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), 72–83, 2018, doi:10.1109/SANER.2018.8330198.

[18] S. Feng, C. Chen, "Prompting Is All You Need: Automated Android Bug Replay with Large Language Models," in Proceedings of the 46th IEEE/ACM International Conference on Software Engineering, 1–13, 2024, doi:10.1145/3597503.3608137.

[19] B. Burg, R. Bailey, A. J. Ko, M. D. Ernst, "Interactive record/replay for web application debugging," in Proceedings of the 26th annual ACM symposium on User interface software and technology, 473–484, 2013, doi:10.1145/2501988.2502050.

[20] J. Hibschman, H. Zhang, "Unravel: Rapid web application reverse engineering via interaction recording, source tracing, and library detection," in Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, 270–279, 2015, doi:10.1145/2807442.2807468.

[21] J. Johnson, J. Mahmud, T. Wendland, K. Moran, J. Rubin, M. Fazzini, "An Empirical Investigation into the Reproduction of Bug Reports for Android Apps," in 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 321–322, 2022, doi:10.1109/SANER53432.2022.00048.

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 9, No. 5, 9-20 (2024)
www.astesj.com
Special Issue on Computing, Engineering and Multidisciplinary Sciences

ASTES Journal
ISSN: 2415-6698

Advancements in Explainable Artificial Intelligence for Enhanced Transparency and Interpretability across Business Applications

Maikel Leon[*,1], Hanna DeSimone[2]

[1]*Department of Business Technology, Miami Herbert Business School, University of Miami, Miami, Florida, USA*

[2]*Miami Herbert Business School, University of Miami, Miami, Florida, USA*

ARTICLE INFO

Article history:
Received: 23 July, 2024
Revised: 14 September, 2024
Accepted: 15 September, 2024
Online: 20 September, 2024

Keywords:
Explainable AI
Transparency
Interpretability

ABSTRACT

This manuscript offers an in-depth analysis of Explainable Artificial Intelligence (XAI), emphasizing its crucial role in developing transparent and ethically compliant AI systems. It traces AI's evolution from basic algorithms to complex systems capable of autonomous decisions with self-explanation. The paper distinguishes between explainability—making AI decision processes understandable to humans—and interpretability, which provides coherent reasons behind these decisions. We explore advanced explanation methodologies, including feature attribution, example-based methods, and rule extraction technologies, emphasizing their importance in high-stakes domains like healthcare and finance. The study also reviews the current regulatory frameworks governing XAI, assessing their effectiveness in keeping pace with AI innovation and societal expectations. For example, rule extraction from artificial neural networks (ANNs) involves deriving explicit, human-understandable rules from complex models to mimic explainability, thereby making the decision-making process of ANNs transparent and accessible. Concluding, the paper forecasts future directions for XAI research and regulation, advocating for innovative and ethically sound advancements. This work enhances the dialogue on responsible AI and establishes a foundation for future research and policy in XAI.

1. Introduction

Artificial Intelligence (AI) has become an increasingly popular topic in recent years. AI is defined as the capability of a machine to replicate cognitive functions associated with the human mind [1]. As new technologies like ChatGPT emerge, uncertainty about the impact of AI technologies on the business world is steadily growing. The complexity of these systems makes it difficult to understand how AI arrives at its conclusions, resulting in a "black box" scenario where the process used to come to a system output is not fully transparent [2]. The black box syndrome in such systems can create problems in critical fields like finance and medical applications. These fields require more transparency and trust when diagnosing or approving a loan. As the use of AI grows, the demand for explainability within knowledge-based systems increases. In the business community, there is worry about human trust in AI recommendations, leading to a desire for transparency in AI systems [3]. The current lack of transparency in AI systems has led to increased focus on the research of explainable AI. There is a clear need for explainability, trust, and transparency in algorithms across various applications. The concept of Explainable AI generalizes new possibilities for AI programs.

The surge in the adoption of AI systems across various sectors necessitates a parallel increase in explainability to ensure these systems are trustworthy, ethical, and accessible. Here are some concise reasons:

- Regulatory Compliance: Increasing global regulations around data privacy and AI transparency demand mechanisms for explaining and justifying automated decisions, especially in critical sectors like healthcare, finance, and legal.
- Ethical Considerations: As AI systems become more prevalent, the ethical implications of their decisions become more significant. Explainable AI facilitates the understanding of automated decisions, supporting ethical auditing and accountability.
- User Trust: Transparency in AI operations fosters user trust and acceptance, crucial for the widespread deployment of AI technologies in sensitive and impactful areas.

These points underscore the essential role of explainability in the responsible scaling of AI technologies. As we delve deeper into the nuances of AI applications, the complexity of these systems grows [4], highlighting the urgent need for advanced research

[*]Corresponding Author: Maikel Leon (mleon@miami.edu)

in explainable AI. Such research not only aids in aligning AI systems with human values and norms but also opens new avenues for innovation in AI governance and policy-making.

2. AI's history highlights

In recent years, the explainability of systems has emerged as a significant factor in adopting AI. It has become essential for practical, social, and legal reasons that users are provided with an explanation of how a system reaches a particular output [5]. Explanations are necessary to understand a system's functions and give users insight into debugging system issues. However, experts have not defined a reason or the qualities it must possess [5]. In early forms of AI, explainability was not prioritized. The origin of AI can be traced back to the 1940s. These first roots of AI were found in the WWII code-breaking machine developed by English mathematician Alan Turing [6]. The technology's ability to outperform humans in decoding caused Turing to question the system's intelligence. In 1950, Turing released a paper discussing how to produce intelligent systems and test their intelligence. In summary, he proposed a test that considered a machine intelligent if a human cannot distinguish between another human and the machine [6]. Today, the Turing Test is still utilized as a benchmark for recognizing the intelligence of a system.

AI foundation traces back to 1956 at Dartmouth College, which kick-started a new era of machine learning research and development. The first hint of explainability can be found in early knowledge-based expert systems in the 1960s. Rule-based expert systems utilize expert human knowledge to solve problems that usually require human-level intelligence [7]. Using expert or domain knowledge, these software systems assist humans in decision-making. Expert systems use an approach of "if-then" statements and have several essential parts, including a knowledge base (usually formatted as a set of rules), an inference engine, and an interface to convey information to a user [6, 7]. Using a top-down approach, expert systems can quickly formalize human intelligence into logical rules that can be followed step-by-step. 1966, at MIT, Joseph Weizenbaum created ELIZA, a natural language processing tool capable of conversing with a human user [6]. ELIZA was one of the first programs to pass the Turing Test. In the early 1970s, governments began to hesitate and pull back funding for AI research, causing a gap in the development of AI.

In the 1980s, Expert Systems, using AI-derived symbolic reasoning techniques to address complex problems, began demonstrating the technology's ability to achieve a firm's goals [8]. However, critics began to argue that overall, expert systems rarely achieved their set goals and, in many cases, could not achieve expert-level performance [8, 9]. These concerns heavily came from the financial sector, as Wall Street did not trust the technology that rarely delivered on its promises.

Due to this suspicion, there was a significant lack of progress in AI initially. There remained a large gap between the expectations and reality of AI capabilities. Expert systems showed impressive potential when attempting problems that can be seamlessly formalized [10]. For example, in 1997, Deep Blue, IBM's chess-playing program, successfully beat Gary Kasparov, the world chess champion, utilizing a tree-search method to evaluate over 200 million potential moves per second [6]. However, this program could not be successfully applied to a problem that is not as quickly standardized, such as face recognition. For a program to accomplish a task like this, the system must correctly interpret data, learn from it, and apply it to various tasks and goals with flexible adaptation [6].

The need for complex decision-making caused an uproar in AI research. While a few machine learning models are labeled interpretable by design - examples include decision trees, rules, and tables- most AI models function as black boxes, meaning the systems do not reveal sufficient details regarding their internal behavior. [5]. The nature of these opaque decision models will be further discussed in the following section. As AI increasingly intertwines with more human-centric applications, the focus has shifted from accuracy to explainability [11].

In the nascent AI development stages, the primary focus was predominantly on enhancing the accuracy and efficiency of AI models:

- Performance Metrics: Early AI research prioritized performance metrics such as precision and recall, with less consideration for how decisions were made within the model.
- Technological Limitations: Limited by the technology of their times, early developers often had to choose between complex, opaque models that offered better performance and simpler, interpretable ones that did not scale.

While this approach was justified in the early days of AI, when the goal was to establish viable, functional AI systems, today's landscape demands a different paradigm [12]. As AI systems increasingly interact with societal and individual decisions, transparency becomes as critical as accuracy. This shift necessitates a robust exploration of XAI, where understanding and clarifying AI processes are not just an academic interest but a societal imperative [13]. The upcoming sections of this paper will delve into the methodologies and impacts of XAI, seeking to bridge the gap between AI capabilities and human-centric values.

3. What Is XAI?

The field of XAI refers to a wide variety of algorithms. These varying algorithms can be grouped by complexity into three main groups: white, gray, and black box models [14]. White-box models are considered systems with full transparency that do not require extra explainability techniques, such as linear regression [14]. Systems that achieve a more advanced performance but lack interpretability, such as neural networks and random forests, are considered black-box models with high accuracy yet lack transparency [11, 14]. These black boxes are considered opaque models, concealing the methods and algorithms mapping inputs to outputs [15]. For example, an opaque system could emerge when an organization licenses closed-source AI to protect its intellectual property and proprietary AI [15]. The "how" and "why" of the system's process are omitted from the output. Finally, gray-box models fall in between, as they are not intrinsically explainable but can be interpretable when explanation techniques are applied [14].

According to the National Institute of Standards and Technology [16], for a system to be considered explainable, it must possess four fundamental properties:

- Explanation: A system must provide accompanying support or evidence with each decision output.
- Meaningful: The system's explanations are understandable to its intended user, considering different user groups' varying knowledge levels and needs.
- Explanation Accuracy: The system's explanation correctly reflects system processes.
- Knowledge Limits: A system only functions within the range of scenarios and conditions it has been trained for. The system can recognize cases that fall outside its scope.

Knowing what "explainability" means is crucial to understanding the importance of explainable AI. The term does not possess an official definition, but experts have culminated several ways to view the concept of explainability. Explainability describes the type of information provided to users through the user interface to allow informed use of a system's output or recommendation [17]. Explainability answers the simple question, "Why did it do that?".

3.1. Explainability, Interpretability, and Transparency

In many cases, explainability and interpretability are used synonymously; however, according to literature on the topic, interpretability and explainability differ slightly. According to Johnson (2020) and Angelov (2021), the definitions of the terms are as follows:

- Explainability: Relates to the concept of explanation as an interface between AI and humans, including AI systems that are comprehensive to humans through explanation [11].
- Interpretability: The ability to determine cause and effect from a machine learning model that is intrinsically understandable to humans [11, 18].

There are notable qualities that explainable and interpretable systems do and do not possess. The terms used are defined as such:

1. Transparency: The quality of AI systems being understandable by themselves, allowing users to comprehend how the system works [11, 19, 20].
2. User Understanding: the ability of human users to immediately make sense of a system's reasoning and behavior without extra explanations or clarifications [21].
3. Comprehensibility: refers to the capacity of a system or a system's explanations to aid a user in task completion [21].
4. Fairness: The goal that explanations should be egalitarian [21].

Systems can be explainable without being interpretable. Explainability considers explanations of the interface between users and an AI system [11]. Explainability is found in AI systems that are accurate and understandable to humans [11]. In addition, explainability works to clarify its internal decision process to users. It emphasizes the ability of parameters, often hidden in deep neural networks, to justify the results [18, 22]. On the other hand, interpretability relates to how accurately a system can link each cause to an effect [18]. Interpretability describes the capacity of a system to give interpretations in formats understandable to humans. Interpretability also includes to what degree users can understand explanations [23]. For example, deep learning models, such as neural networks, tend to perform highly but lack interpretability [14, 24].

In both interpretable and explainable AI systems, fairness is not guaranteed. Although these techniques provide insight into model behavior and reveal biases, achieving fairness requires the consideration of factors including data bias, algorithmic fairness, and ethical considerations [20]. A system's explainability can be determined by several factors, including complexity, transparency, trust, fidelity, accuracy, and comprehensibility [5, 16, 23]. These dimensions of explainability distinguish explainable systems from black-box models and are critical pieces of explainable AI.

One necessary element of explainable AI is transparency. While explainability answers the question "Why did it do that?" transparency addresses "How does it work?" [25]. In summary, transparency is found in systems that have the potential to be understandable by themselves, making transparent systems the opposite of black box models [11]. Transparency helps lift the lid of black box models. This can reveal a model's structural attributes, evaluation metrics, and descriptive properties from training data to users to foster an understanding of a system's underlying logic [5, 25]. Many machine learning models lacked transparency due to a trade-off between explainability and performance [19]. As previous studies focused on performance improvement, transparency was ignored and placed on the back burner.

AI systems' nontransparent nature began to affect human trust and confidence negatively. More specialized knowledge became necessary to understand AI approaches as the complexity increased. Ordinary users with low algorithmic knowledge found it hard to trust AI systems making crucial decisions, and the lack of transparency hindered user understanding of the exact steps of algorithms [26]. This significantly worsened the problem, as user comprehension of why a specific recommendation is made and how their input affects the results is critical to user satisfaction and trust [26]. For example, in a news recommender system, fair and personalized recommendations give users confidence, leading to trust and continued use [26]. Visible transparency improves search performance, as using explanations improves users' overall satisfaction [26]. In recommender systems, personalization has become a determinant of satisfaction and trust [26]. Moreover, the recommendation explanation sets a prerequisite for a relationship of trust between humans and AI [2]. A lack of transparency in medical applications has been identified as a barrier to AI implementation [23]. Trust in medical AI systems is vital, as the recommendations significantly impact patients' health and well-being [23].

The need for transparency has led to a significant interest in XAI. This field ensures that AI benefits rather than harms society by introducing accountability [3]. Systems that lack transparency don't possess this accountability. In some cases, this is not an issue. For instance, in the historic Go game between Lee Sodel, a highly skilled Go player, and AlphaGo, a DeepMind AI system, AlphaGo made an extremely unexpected move [2]. Experts were unsure why the system made this gaming-altering move. In this case, the nontransparent nature of AlphaGo did not matter, as the

application did not drastically affect human well-being. However, in many applications, the opposite is true.

On the other hand, IBM Watson, a supercomputer containing AI and other analytical software, beat the top players at the game show Jeopardy. This software was then marketed to medical facilities as a cancer-detecting system [2]. When providing results, Watson could not display the reasoning for its output, so patients and doctors could not trust the system [2, 3]. IBM Watson's lack of transparency hindered human trust and was not seen as a successful application. The same mindset can be applied to self-driving cars as well. These automated systems did not react efficiently in a new or unfamiliar environment. In 2018, a computerized vehicle owned by Uber crashed, and the operator was charged with negligent homicide [11]. Transparent and explainable systems are necessary, from a public trust perspective and a legal viewpoint, to provide more reliable and safe systems [11].

The capacity of AI-based systems to elucidate their internal decision-making processes is an area ripe for exploration and innovation:

- Model Transparency: Techniques such as model visualization and feature importance metrics provide insights into the working of complex models, enhancing their transparency.
- Decision Justification: Implementing methods that allow AI systems to justify their decisions can facilitate greater understanding and trust among users.

As AI technologies continue to permeate various aspects of personal and professional life, the ability of these systems to offer clear, understandable explanations for their actions becomes crucial. This supports the development of more robust and reliable AI and upholds the user's right to demand transparency [27]. The next section of this paper will discuss methodologies for formulating these explanations, ensuring that AI systems are effective, accountable, and accessible to the users they serve.

4. Explanations

According to [5], explanations can be understood in two ways: as a line of reasoning or as a problem-solving activity. Viewing explanations as a line of reasoning essentially creates understanding by following the path inference rules take to come to a particular decision [5]. The main issue with this approach was the complexity of explanations, as not all users possess the same knowledge to understand the full extent of explanations thoroughly. This idea was re-conceptualized to approach explanation in a different light: explanations as a problem-solving activity. This altered view not only reconstructs the system's reasoning but also considers various degrees of abstraction, meaning different knowledge levels were considered [28].

Post-hoc and model-based explanations are the most prevalent types when categorizing the explanations provided by XAI systems. Post-hoc methods are commonly used on systems that are not intrinsically interpretable to boost their interpretability [29]. Post-hoc methods do not directly reveal the internal workings of a model. Still, they seek to explain behavior to users by studying outputs and factors that contribute to the result [16]. In other words, explanations are derived after a model makes the prediction. The system uses the nature and attributes of results to generate explanations [17].

On the other hand, model-based explanations focus on the mechanical aspect of recommendations and aim to illustrate how an algorithm suggests a distinct output [5]. Model-based explanation strategies use a different model to explain how the task model functions. The levels of soundness and fidelity are particularly essential for assessing model-based explanations [23]. Model-based explanations are strictly based on the system's underlying assumptions and structure [5]. The following subsections briefly overview post-hoc explanations, addressing different techniques and applications. In addition, several other relevant explanation types, such as self-interpretable models, are referenced.

4.1. Post-hoc Explanations

Post-hoc explainability can be applied in two ways: model-specific and model-agnostic approaches. Model-specific methods produce explanations by utilizing the particular system's internal learning process [30]. Since model-specific interpretability is tailored to bring transparency to specific models, the application will not be suitable for other model types [11, 20, 30]. In contrast, model-agnostic methods are independent of the applied system. Model-agnostic methods develop end-user explanations using the inputs and predictions of the model [20, 30]. The lack of specificity of model-agnostic methods allows for wide-scale usage. In addition, the interpretability of post-hoc models can be further divided into local and global methods.

4.1.1. Local Methods

Local methods obtain explainability by segmenting the solution space and providing less intricate explanations that apply to the entire model [29]. A per-decision or single-decision explanation is the most dominant type of local explanation [16]. It provides insight into the aspects that impact the algorithm's decision for a particular input. Local explanations allow for a local approximation of how a black-box model functions [11]. The most well-known example of local methods is LIME (Local Interpretable Model Agnostic Explainer) [16, 17]. LIME functions by taking a decision and creating an interpretable model that illustrates the local decision, which is then used to deliver per-feature explanations [16]. LIME perturbs training data into a new dataset to form a new interpretable model [11]. Another example of local explanations is SHAP (Shapely Additive exPlanations), which uses a mechanism of additive feature attributions to reveal the significance of input factors [14, 17].

4.1.2. Global Methods

Global methods employ interpretable mechanisms, such as decision trees, to extract a simplified version of a complex black box model to supply understandable explanations for each decision made by the model [11]. This makes it possible to comprehend the behavior of the black-box model and how it relates to its trained characteristics [11]. Global explanations can construct post-hoc explanations

on the whole algorithm [16]. Partial Dependence Plots (PDPs) and Testing with Concept Activation Vectors (TCAV) are examples of global explanations. PDPs demonstrate the modification of predicted responses about altered data components. At the same time, TCAVs explain deep neural networks in a more user-friendly manner and have been applied to image classification systems [16].In addition, a global variant of LIME exists, SP-LIME, which uses applicable local LIME outputs as synopsis explanations [16].

4.2. Self-Interpretable Models

Self-interpretable models are intrinsically explainable, meaning humans can directly understand them. The models are the explanation due to a transparent reasoning process [16, 31, 32]. However, many sources claim self-interpretable models are less accurate than post-hoc explanations due to a trade-off between accuracy and interpretability [16, 33]. The most common self-interpretable models include regression models and decision trees [16, 34].

4.3. Other Explanation Models

In addition, several other explanations exist that do not perfectly fit into a category. The most relevant of these explanation models are defined below.

Forms of Model Explanations:

- Introspective Methods: Explanations are formed by connecting inputs to outputs in black-box models. For example, reflective methods can be applied to image classifications with Deep Neural Networks [5, 35, 36] and [37].
- Counterfactual Methods: Explanations provide "what-if" statements regarding how the outputs of a predicted model could be affected by input changes [5, 38, 39, 40] and [41].
- Explanation by Feature Relevance: A method of post-hoc explainability clarifies a model's internal functioning by calculating a relevance score for each variable. The comparison of scores depicts the weight each variable holds [20, 42] and [43].
- Explanation by Simplification: Explanations that use a trained model to formulate a simplified representation to assemble an easily implementable model. These models optimize similarity to the original model while simultaneously decreasing complexity [11, 29] and [44].

AI-based systems must explain their decisions, which may soon transition from a best practice to a mandatory requirement. This shift is driven both by evolving regulatory frameworks aimed at safeguarding consumer rights and by ethical standards that promote transparency and accountability [45]:

- Regulatory Compliance: Legislations such as the EU's General Data Protection Regulation (GDPR) already impose obligations on AI to explain decisions that affect individuals, signaling a broader trend towards legal mandates.
- Ethical Accountability: Beyond compliance, there is a growing recognition of the ethical obligation for AI to be transparent, particularly in systems that impact public welfare and individual freedoms.

This development is poised to significantly benefit numerous business sectors by enhancing consumer trust, facilitating more informed decision-making, and improving the overall user experience with AI technologies.

5. From ANNs (sub-symbolic) to Rules (symbolic)

Extracting rules from ANNs is crucial in demystifying these models' "black-box" nature, making their decisions understandable and interpretable to humans. This process involves translating the intricate, non-linear relationships learned by the network into a set of rules that humans can easily understand. To illustrate this process, we'll explore a detailed example of how rules can be extracted from an ANN trained on a simplified dataset for predicting loan approval based on applicant features.

5.1. Background

Let us use the example of a fictional financial institution that has created an ANN to evaluate loan applications. The ANN considers various applicant features such as Age, Income, Credit Score, and Employment Status and provides a binary decision: Approve or Deny. Despite the ANN's high accuracy, the decision-making process is not transparent. This makes it challenging for loan officers to explain decisions to applicants or to ensure compliance with regulations. The institution aims to derive understandable rules from the ANN to address this.

5.2. ANN Architecture

The ANN in this example is a simple feedforward network with one hidden layer. The input layer has four neurons corresponding to the applicant features. The hidden layer has a few neurons (say five for simplicity) using ReLU (Rectified Linear Unit) as the activation function [41]. The output layer has one neuron and uses a sigmoid activation function to output a probability of loan approval.

5.3. Rule Extraction Process

The rule extraction process involves several steps designed to translate the ANN's learned weights and biases into a set of if-then rules that replicate the network's decision-making process as closely as possible:

- Simplification: The first step involves simplifying the ANN to make the rule extraction more manageable. This could include pruning insignificant weights (shallows values) and neurons that have little impact on the output based on sensitivity analysis.
- Discretization: Since ANNs deal with continuous inputs and hidden layer activations, a discretization process is applied to convert these continuous values into categorical ranges. For

instance, age might be categorized into 'Young', 'Middle-aged', and 'Old'; Income into 'Low', 'Medium', and 'High'; Credit Score into 'Poor', 'Fair', 'Good', and 'Excellent'; and Employment Status into 'Unemployed' and 'Employed'.

- Activation Pattern Analysis: Next, the activation patterns of the neurons in the hidden layer are analyzed for each input pattern. This involves feeding various combinations of the discretized input variables into the simplified network and observing which neurons in the hidden layer are activated for each combination. An activation threshold is defined to determine whether a neuron is considered activated.

- Rule Generation: Based on the activation patterns observed, rules are generated to replicate the ANN's decision process. Each rule corresponds to a path from the input layer through the activated hidden neurons to the output decision. For example:

  - If (Age is Young) and (Income is High) and (Credit Score is Good) and (Employment Status is Employed), then Approve Loan.
  - If (Age is Middle-aged) and (Credit Score is Poor), then Deny Loan.

  This step involves identifying which combinations of input features and hidden neuron activations lead to loan approval or denial, effectively translating the ANN's complex decision boundaries into more interpretable formats.

- Rule Refinement and Validation: The initial set of rules may be too complex or too numerous for practical use. Rule refinement techniques simplify and consolidate the rules without significantly reducing their accuracy in replicating the ANN's decisions. The refined rules are then validated against a test dataset to accurately reflect the ANN's behavior. This may involve adjusting the rules based on misclassifications or applying techniques to handle exceptions and edge cases.

After applying the rule extraction process to our hypothetical ANN, we might end up with a set of simplified, human-readable rules such as:

- Rule 1: If (Income is High) and (Credit Score is Excellent), then Approve Loan.
- Rule 2: If (Employment Status is Unemployed) and (Credit Score is Poor or Fair), then Deny Loan.
- Rule 3: If (Age is Old) and (Income is Low) and (Employment Status is Employed), then Deny Loan.

These rules provide clear criteria derived from the ANN's learned patterns, making the decision-making process transparent and justifiable.

5.4. Advantages and Challenges

Some advantages include:

- Transparency: The extracted rules make the ANN's decisions transparent and understandable to humans.
- Compliance: Clear rules can help ensure compliance with regulatory requirements for explainable AI.
- Trust: Understanding how decisions are made can increase user trust in the AI system.

Some challenges are:

- Complexity: The rule extraction process can be complex, especially for deep or highly non-linear networks [46].
- Approximation: The extracted rules approximate the ANN's decision process and may not capture all nuances.
- Scalability: Extracting rules from large, deep neural networks with many inputs and hidden layers can be challenging and may result in many complex rules [47].

5.5. Summary

Extracting rules from ANNs makes AI decision-making transparent, understandable, and justifiable. Although there are challenges, especially with complex networks, this process is crucial for responsible and ethical AI use. By making AI systems more interpretable, we can establish trust with users, ensure compliance with regulations, and gain valuable insights into decision-making.

6. Fuzzy Cognitive Maps

The pendulum in AI is swinging back from purely statistical approaches toward integrating structured knowledge. FCMs are powerful cognitive tools for modeling and simulating complex systems. They blend elements from artificial neural networks, graph theory, and semantic nets to offer a unique approach to understanding and predicting system behavior. FCMs incorporate the concept of fuzziness from fuzzy logic, enabling them to handle ambiguity and uncertainty inherent in real-world scenarios. This extensive report delves into the origins of FCMs, provides illustrative case studies, and discusses their advantages and disadvantages, with references to their similarities to artificial neural networks, graphs, and semantic nets [48].

6.1. Origins

Bart Kosko introduced the concept of FCMs in the 1980s as an extension of cognitive maps. Cognitive maps, developed by Axelrod, were diagrams that represented beliefs and their interconnections. Kosko's introduction of fuzziness to these maps allowed for the representation of causal reasoning with degrees of truth rather than binary true/false values, thus capturing the uncertain and imprecise nature of human knowledge and decision-making processes. FCMs combine elements from fuzzy logic, introduced by Lotfi A. Zadeh, with the structure of cognitive maps to model complex systems.

6.2. Structure and Functionality

FCMs are graph-based representations where nodes represent concepts or entities within a system, and directed edges depict the causal relationships between these concepts. Each edge is assigned a weight that indicates the relationship's strength and direction (positive or negative). This structure closely mirrors that of artificial neural networks, particularly in how information flows through the network and how activation levels of concepts are updated based on the input they receive, akin to the weighted connections between neurons in neural networks [49].

However, unlike typical neural networks that learn from data through backpropagation or other learning algorithms, the weights in FCMs are often determined by experts or derived from data using specific algorithms designed for FCMs. The concepts in FCMs can be activated like neurons, with their states updated based on fuzzy causal relations, allowing for dynamic modeling of system behavior over time. Integrating structured knowledge graphs with distributed neural network representations offers a promising path to augmented intelligence. We get the flexible statistical power of neural networks that predict, classify, and generate based on patterns—combined with the formalized curated knowledge encoding facts, logic, and semantics via knowledge graphs [50].

6.3. The Inherent Reasoning Mechanism

The primary function of the reasoning rule in FCM models is to update the activation values of concepts iteratively, starting from initial conditions and continuing until a stopping criterion is satisfied. During each iteration, the reasoning rule utilizes three primary components to conduct these calculations: the weight matrix, which signifies the connections between concepts; the activation values of concepts from the previous iteration; and the activation function.

Eq. (1) shows a general rule commonly found in FCMs-related papers:

$$a_i^{(t)} = f\left(\sum_{j=1, i\neq j}^{N} a_j^{(t-1)} w_{ji}\right), \tag{1}$$

Recently, in [51], the author proposed an updated quasi-nonlinear reasoning rule depicted in Eq. (2):

$$a_i^{(t)} = \underbrace{\phi \cdot f\left(\sum_{j=1}^{N} a_j^{(t-1)} w_{ji}\right)}_{\text{nonlinear component}} + \underbrace{(1-\phi)\cdot a_i^{(0)}}_{\text{linear component}}, \tag{2}$$

such that $0 \leq \phi \leq 1$ is the nonlinearity coefficient. When $\phi = 1$, the concept's activation value depends on the activation values of connected concepts in the previous iteration. When $0 < \phi < 1$, we add a linear component to the reasoning rule devoted to preserving the initial activation values of concepts. When $\phi = 0$, the model narrows down to a linear regression where the initial activation values of concepts act as regressors. In their paper, Nápoles et al. [51] used the quasi-nonlinear reasoning rule to quantify implicit bias in pattern classification datasets. In contrast, the authors in [41] resorted to this rule to develop a recurrence-aware FCM-based classifier.

6.4. How Activation Functions Work

The activation function $f : \mathbb{R} \rightarrow I$ is an essential component in the reasoning rule of FCM-based models. This monotonically non-decreasing function keeps the activation value of each concept within the desired image set I, which can be discrete (a finite set) or continuous (a numeric-valued interval). It should be mentioned that I must be bounded; otherwise, the reasoning rule could explode due to the successive additions and multiplications when updating concepts' activation values during reasoning. Table **??** portrays relevant activation functions found in the literature.

6.5. Relevant Case Studies

For illustration purposes, Figure 1 shows an example of an FCM created to model a case of autism [32]. FCMs have been applied across various domains, demonstrating their versatility and effectiveness as a hybrid AI tool:

- Decision Support Systems: FCMs model complex decision-making processes, integrating expert knowledge and data-driven insights to support decisions in healthcare, environmental management, and business strategy.

- Predictive Modeling: In healthcare, FCMs model the progression of diseases or the impact of treatments, incorporating medical expertise and patient data to predict outcomes and support personalized medicine [52].

- System Analysis and Design: FCMs help analyze and design complex systems, such as socio-economic systems or ecosystems, by modeling the interactions between various factors and predicting the impact of changes or interventions.

- Healthcare Management: FCMs have been employed to model and predict patient outcomes in healthcare settings. For example, an FCM can be developed to understand the complex interplay between patient symptoms, treatment options, and possible outcomes, aiding medical professionals in decision-making [53].

- Environmental and Ecological Systems: In environmental studies, FCMs have been used to model the impact of human activities on ecosystems, allowing for the simulation of various scenarios based on different policies or interventions. This application showcases the strength of FCMs in handling systems where data may be scarce or imprecise [54].

- Business and Strategic Planning: FCMs assist in strategic planning and decision-making within business contexts by modeling the relationships between market forces, company policies, and financial outcomes, offering a tool for scenario analysis and strategy development [55].

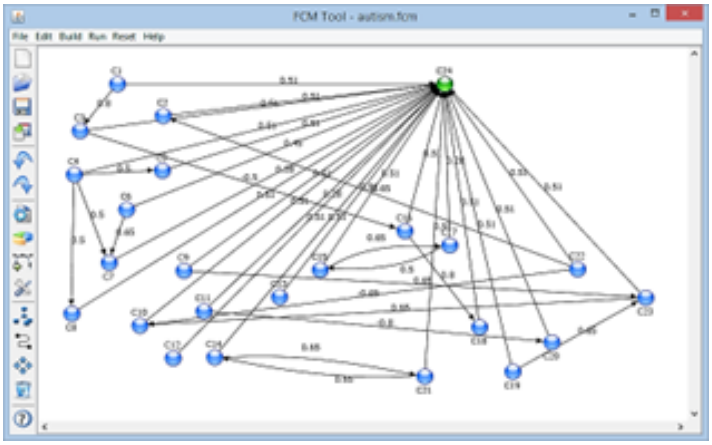


Figure 1: Real example created with FCM Tool.

6.6. Advantages

The hybrid nature of FCMs offers several advantages:

- Interpretability and Transparency: The symbolic representation of concepts and causal relationships in FCMs provides clarity and understandability, facilitating communication with experts and stakeholders and supporting explainable AI.
- Flexibility and Adaptability: FCMs can be easily updated with new knowledge or data, allowing them to adapt to changing conditions or insights. This makes them particularly valuable in fields where knowledge evolves rapidly.
- Handling of Uncertainty: Using fuzzy values to represent causal strengths enables FCMs to deal effectively with uncertainty and ambiguity, providing more nuanced and realistic modeling of complex systems [4].
- Integration of Expert Knowledge and Data-Driven Insights: FCMs uniquely combine expert domain knowledge with learning from data, bridging the gap between purely knowledge-driven and purely data-driven approaches.
- Interpretability: The graphical representation of FCMs, similar to semantic nets, allows for straightforward interpretation and understanding of the modeled system, making it accessible to experts and stakeholders without deep technical knowledge of AI.
- Flexibility: FCMs can incorporate quantitative and qualitative data, effectively handling uncertainty and imprecision through fuzzy logic. This flexibility makes them suitable for a wide range of applications.
- Dynamic Modeling Capability: FCMs can simulate the dynamic behavior of systems over time, providing valuable insights into potential future states based on different inputs or changes in the system [56].

6.7. Limitations

Despite their advantages, FCMs also face several challenges:

- Complexity with Large Maps: As the number of concepts and relationships in an FCM increases, the map can become complex and challenging to manage, analyze, and interpret [57].
- Learning and Optimization: While FCMs can learn from data, adjusting the fuzzy values of causal relationships can be computationally intensive and require sophisticated optimization techniques, especially for large and complex maps [58].
- Quantification of Expert Knowledge: Translating expert knowledge into precise fuzzy values for causal relationships can be challenging and may introduce subjectivity, requiring careful validation and sensitivity analysis [59].
- Subjectivity in Model Construction: The reliance on expert knowledge for constructing FCMs can introduce subjectivity, especially in determining the strength and direction of causal relationships between concepts.
- Complexity with Large Maps: As the number of concepts increases, the FCM can become complex and challenging to manage and interpret, potentially requiring sophisticated computational tools for simulation and analysis.
- Limited Learning Capability: While FCMs can be adjusted or trained based on data to some extent, they lack the deep learning capabilities of more advanced neural networks, which can autonomously learn complex patterns from large datasets [60].

7. Applications

Numerous potential applications exist for XAI techniques and models, including healthcare, law, data science, and business [55]. This section explores the need for explainability in these applications, including their current uses, limitations, and future development.

7.1. Healthcare

In healthcare, there are many applications of XAI such as diagnosis, treatment recommendations, and surgery [23, 61, 62]. For example, an explainable model was proposed for diagnosing skin diseases. Using saliency maps to highlight important parts of the image crucial to diagnosis, dermatologists can easily understand the model's arrival at a diagnosis and then provide a more in-detail diagnosis [61]. According to a survey by [62], LIME is the most commonly used XAI approach in medical applications.

Throughout the COVID-19 pandemic, AI has shown potential in developing solutions to confront the difficulties presented by the virus [61]. However, the lack of transparency in black-box models has hindered their acceptance in clinical practice. With the development in user trust and model performance, XAI can attempt these problems in the future [61]. XAI techniques have been created in the context of medical image analysis to facilitate disease detection and diagnosis through feature visualization [61]. This allows medical professionals and their patients to obtain a deeper insight into the model's process, building confidence in its accuracy. In high-stakes applications, specifically healthcare, there is debate about whether

explainable modeling is necessary. To some, explainability is crucial. On the other hand, some say prioritizing explainability above accuracy in healthcare systems can be unethical [23]. According to [23], the post-hoc explanations can be delusive, but a potential solution is to create post-hoc explanation models with argumentative support.

Suppose the case of an ANN equipped with a rule extraction method can be deployed to diagnose diseases from medical imaging with high accuracy. The ANN processes complex imaging data to identify patterns indicative of specific conditions, such as tumors in MRI scans. A rule extraction technique is integrated into the system to ensure clinicians and patients understand the diagnostic process. This technique translates the ANN's intricate decision-making into simple, interpretable rules, such as the presence of specific shapes or textures associated with malignancy. This not only aids medical professionals in making informed treatment decisions but also enhances patient trust by providing clear explanations for the diagnoses made by the AI system.

7.2. Law

In the context of legal applications, XAI possesses several potential applications. As stated by [61], XAI can be used for legal document analysis, contract review, legal decision-making, and addressing challenges in legal domains. AI can help analyze large volumes of legal documents and sort significant information to facilitate a more accurate analysis, as well as assist in recommending plea bargains or predicting case outcomes [61, 63]. Despite the increasing emphasis on AI in the legal world, systems still struggle to perform at necessary levels due to the precise nature of legal work. Such characteristics include the exact nature of legal jargon, the high level of expertise required, the mass amount of situational exceptions, and the limited tolerance of mistakes [61]. The motivation for interpretable, explainable, and trustworthy systems feeds the recent upsurge of XAI research in legal applications.

In legal applications, an FCM can be a sophisticated tool for modeling and visualizing the intricate dynamics of legal cases and legislative processes. By capturing and representing the causal relationships between various legal factors—such as statutes, precedents, and evidentiary variables—FCMs enable legal professionals to simulate and scrutinize the potential outcomes of different legal strategies in a visually interpretable format. This capability goes beyond basic explainability by showing outcomes and allowing users to interact with the map to adjust variables and immediately see different scenario outcomes. This interactive, interpretable visualization aids in understanding complex legal interdependencies, facilitating more informed decision-making and strategy formulation, especially in cases involving overlapping laws and diverse outcomes.

7.3. Finance

In the financial sector, the applications of XAI can be split into thematic categories. These clusters include financial distress and corporate failure, algorithmic and high-frequency trading, forecasting/predictive analysis, text mining and sentiment analysis, financial fraud, pricing and valuation, scheduling, and investor behavior [64]. In addition, [61] describe the potential applications of XAI in finance as follows:

- **Fraud Detection:** Explain decisions by identifying the reasons behind fraudulent activities and prevent future issues.
- **Credit Scoring:** Allows banks and their customers to understand exactly why a particular credit score was calculated and facilitates lending decisions.
- **Investment Management:** Increased transparency in portfolio management can lead to better performance and more satisfied investors.
- **Compliance:** XAI could assist in mitigating potential biases and avoiding legal issues.
- **Customer Service:** XAI will improve customer service by, for example, including explanations along with loan denials to improve customer understanding and satisfaction.

According to additional literature on the topic, subjects within the finance domain commonly discussed as potential applications of XAI include risk management, portfolio optimization, electronic financial transaction clarification, and anti-money laundering [64]. Due to the high level of regulations in financial domains, XAI is necessary to augment processes to ensure trust and transparency and mitigate risks [65].

Suppose the case of an ANN equipped with a rule extraction method can be effectively used for credit scoring. The ANN analyzes extensive data sets, including transaction history, payment behavior, and credit utilization, to assess the creditworthiness of applicants. By integrating a rule extraction method, the system can transparently generate and provide clear, human-understandable rules that explain its credit-scoring decisions. This transparency not only aids financial analysts in understanding the model's decision-making process but also ensures compliance with regulatory requirements regarding fairness and explainability in credit assessments.

An FCM can model and visualize a client's financial stability or market for the same finance application. By representing elements like market trends, economic indicators, and individual financial behaviors as nodes and their interdependencies as edges, FCMs allow financial analysts to simulate and interpret complex financial scenarios. This method provides a dynamic, interpretable visualization beyond mere explanation, enabling interactive exploration of potential financial outcomes based on varying inputs. Such interpretability is invaluable in strategic financial planning and risk assessment, allowing the decision-makers to foresee and mitigate potential financial instabilities or crises.

8. Future

As complex and human-centric systems become more prevalent, there is a growing need for explainable AI in many applications. Due to the rapid increase in AI, there are currently few regulations and rules governing these systems. However, as the need for trust and transparency continues to rise, regulations are essential to ensure both ethical and accountable AI.

8.1. Current Regulations

Historically, AI-based systems have operated in an environment with minimal regulatory oversight regarding their need to explain internal decision-making processes:

- Early AI Developments: Initially, AI technologies were developed and deployed with a focus on functionality and performance, often at the expense of transparency and accountability [66].

- Regulatory Lag: There has been a significant lag in developing and implementing regulations that require AI systems to be explainable, partly due to the rapid pace of technological advancement outstripping policy development.

However, as the implications of AI technologies have become more apparent, there is a growing consensus among government bodies and policymakers about the necessity of regulatory frameworks that ensure AI systems are transparent and accountable. This shift reflects a broader awareness of AI's potential impacts on society and the need for appropriate safeguards.

The regulation of AI is becoming extremely important in terms of ethics and responsible decision-making. The European Union's General Data Protection Regulation (GDPR) was put into effect in 2018, and the GDPR has raised several legal and ethical questions regarding safety, responsibility, malfunction liability, and the overall trade-offs associated with AI decisions [67]. The GDPR gives citizens a "right to explanation" in algorithmic choices that significantly affect them [68, 69]. Regulations like the GDPR make it nearly impossible to use black-box models in various sectors, emphasizing the growing need for explainability and transparency [70, 71]. Additionally, the EU's intense regulatory actions involving digital markets, including the AI domain, strive to provide an ethical approach to AI applications [72]. Additionally, Hacker (2023) highlights the transformative prospects as well as risks associated with large generative AI models (LGAIMs), such as ChatGPT, and how current regulations are not suited to manage this class of AI [73].

In April 2021, the European Commission proposed a groundbreaking proposal for the first-ever EU regulatory framework for AI. This framework consists of a risk-based classification technique in which the level of risk specifies the regulation applied to a system [74]. The AI Act manages the opacity of particular systems, emphasizing systems classified as high-risk through a focus on transparency [75]. If implemented, the AI Act will represent the world's baseline rules for overseeing AI. Furthermore, generative AI systems such as ChatGPT must follow transparency conditions, such as publishing data synopses for training the system [74].

In summary, AI regulations are developing to address ethical considerations, transparency, and the responsible use of AI across diverse sectors. The GDPR and corresponding endeavors emphasize the demand for transparency and accountability in AI decision-making. At the same time, ongoing discussions in the EU seek to shape AI development in a human-centric and ethical fashion.

8.2. The Future of XAI

The future of XAI holds tremendous promise and challenges. In an increasingly AI-driven world, the possible applications are extensive; however, awareness of the fragile nature and potential biases within AI systems is expanding. As stated previously, global organizations are attempting to craft standards for responsible AI to mitigate concerns. These regulations strive to make AI systems exemplify more transparency and accountability, making the demand for explainable systems higher than ever.

As different organizations and governments pass regulations, the dilemma now shifts: Is regulating the AI available to specific users and not others ethical? When tackling this issue, enforcing rules on AI is essential. Without universal regulations, organizations may pass conflicting laws, which could immensely harm companies attempting to operate systems globally. For example, with search engines experimenting with generative AI systems, such as Google's Bard or Gemini, non-universal regulations would require several system versions to adhere to local regulations, causing unnecessary complexities. Moreover, universal regulations would provide businesses with legal certainty. Ethically, universal regulations will form a standard for ethical AI, assisting in eliminating biased and discriminatory systems. This will also allow users to feel more trust in consistently observed systems, leading to increased adoption of systems.

In conclusion, from a business perspective, the universal enforcement of AI regulations provides many advantages. Companies should prioritize accountable AI and support coordinated regulations to develop ethical, transparent, and innovative AI technologies. Explainable systems are the key to the future of Responsible AI.

References

[1] V. K. Michael Chui, B. McCarthy, "An Executives Guide to AI," McKinsey & Company, 2018.

[2] W. Samek, T. Wiegand, K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," arXiv preprint arXiv:1708.08296, 2017, doi:10.48550/arXiv.1708.08296.

[3] "Unlocking the black box with explainable AI - Infosys," Infosys, 2019.

[4] M. Leon, "Aggregating Procedure for Fuzzy Cognitive Maps," The International FLAIRS Conference Proceedings, **36**(1), 2023, doi:10.32473/flairs.36.133082.

[5] R. Confalonieri, L. Coba, B. Wagner, T. R. Besold, "A historical perspective of explainable Artificial Intelligence," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **11**(1), e1391, 2021, doi:10.1002/widm.1391.

[6] M. Haenlein, A. Kaplan, "A brief history of artificial intelligence: On the past, present, and future of artificial intelligence," California Management Review, **61**(4), 5–14, 2019, doi:10.1177/0008125619864925.

[7] A. Abraham, "Rule-Based expert systems," Handbook of Measuring System Design, 2005, doi:10.1002/9780470027325.s6405.

[8] T. G. Gill, "Early expert systems: Where are they now?" MIS Quarterly, **19**(1), 51–81, 1995, doi:10.2307/249711.

[9] J. Kastner, S. Hong, "A review of expert systems," European Journal of Operational Research, **18**(3), 285–292, 1984, doi:10.1016/0377-2217(84)90202-0.

[10] E. Struble, M. Leon, E. Skordilis, "Intelligent Prevention of DDoS Attacks using Reinforcement Learning and Smart Contracts," The International FLAIRS Conference Proceedings, **37**(1), 2024, doi:10.32473/flairs.37.1.135349.

[11] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, P. M. Atkinson, "Explainable artificial intelligence: an analytical review," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, **11**(5), e1424, 2021, doi:10.1002/widm.1424.

[12] M. Leon, "Fuzzy Cognitive Maps as a Bridge between Symbolic and Subsymbolic Artificial Intelligence," International Journal on Cybernetics & Informatics (IJCI). 3rd International Conference on Artificial Intelligence Advances (AIAD 2024), **13**(4), 57–75, 2024, doi:10.5121/ijci.2024.130406.

[13] M. Leon, L. Mkrtchyan, B. Depaire, D. Ruan, K. Vanhoof, "Learning and clustering of fuzzy cognitive maps for travel behaviour analysis," Knowledge and Information Systems, **39**(2), 435–462, 2013, doi:10.1007/s10115-013-0616-z.

[14] M. Schemmer, N. Kühl, G. Satzger, "Intelligent decision assistance versus automated decision-making: Enhancing knowledge work through explainable artificial intelligence," arXiv preprint arXiv:2109.13827, 2021, doi:10.48550/arXiv.2109.13827.

[15] D. Doran, S. Schulz, T. R. Besold, "What does explainable AI really mean? A new conceptualization of perspectives," arXiv preprint arXiv:1710.00794, 2017, doi:10.48550/arXiv.1710.00794.

[16] P. J. Phillips, C. A. Hahn, P. C. Fontana, D. A. Broniatowski, M. A. Przybocki, "Four principles of explainable artificial intelligence," Gaithersburg, Maryland, 18, 2020, doi:10.6028/NIST.IR.8312.

[17] P. Bhattacharya, N. Ramesh, "Explainable AI: A Practical Perspective," Infosys, 2020.

[18] J. Johnson, "Interpretability vs explainability: The black box of machine learning," BMC Blogs, 2020.

[19] D. Minh, H. X. Wang, Y. F. Li, T. N. Nguyen, "Explainable artificial intelligence: a comprehensive review," Artificial Intelligence Review, **55**, 3503–3568, 2022, doi:10.1007/s10462-021-10088-y.

[20] A. Rai, "Explainable AI: From black box to glass box," Journal of the Academy of Marketing Science, **48**, 137–141, 2020, doi:10.1007/s11747-019-00710-5.

[21] C. Meske, E. Bunde, J. Schneider, M. Gersch, "Explainable Artificial Intelligence: Objectives, Stakeholders and Future Research Opportunities," Information Systems Management, 2020, doi:10.1080/10580530.2020.1849465.

[22] S. S. Band, A. Yarahmadi, C.-C. Hsu, M. Biyari, M. Sookhak, R. Ameri, I. Dehzangi, A. T. Chronopoulos, H.-W. Liang, "Application of explainable artificial intelligence in medical health: A systematic review of interpretability methods," Informatics in Medicine Unlocked, **40**, 101286, 2023, doi:10.1016/j.imu.2023.101286.

[23] A. F. Markus, J. A. Kors, P. R. Rijnbeek, "The role of explainability in creating trustworthy artificial intelligence for health care: a comprehensive survey of the terminology, design choices, and evaluation strategies," Journal of Biomedical Informatics, **113**, 103655, 2021, doi:10.1016/j.jbi.2020.103655.

[24] R. Goebel, A. Chander, K. Holzinger, F. Lecue, Z. Akata, S. Stumpf, P. Kieseberg, A. Holzinger, "Explainable AI: the new 42?" in Machine Learning and Knowledge Extraction: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings 2, 295–303, Springer, 2018, doi:10.1007/978-3-319-99740-7_21.

[25] C. Oxborough, E. Cameron, A. Rao, A. Birchall, A. Townsend, C. Westermann, "Explainable AI: Driving business value through greater understanding," Retrieved from PWC website: https://www.pwc.co.uk/audit-assurance/assets/explainable-ai.pdf, 2018.

[26] D. Shin, "The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI," International Journal of Human-Computer Studies, **146**, 102551, 2021, doi:10.1016/j.ijhcs.2020.102551.

[27] G. Nápoles, M. L. Espinosa, I. Grau, K. Vanhoof, R. Bello, Fuzzy cognitive maps based models for pattern classification: Advances and challenges, volume 360, 83–98, Springer Verlag, 2018.

[28] G. Nápoles, M. Leon, I. Grau, K. Vanhoof, "FCM Expert: Software Tool for Scenario Analysis and Pattern Classification Based on Fuzzy Cognitive Maps," International Journal on Artificial Intelligence Tools, **27**(07), 1860010, 2018, doi:10.1142/S0218213018600102.

[29] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, **58**, 82–115, 2020, doi:10.1016/j.inffus.2019.12.012.

[30] D. Vale, A. El-Sharif, M. Ali, "Explainable artificial intelligence (XAI) post-hoc explainability methods: Risks and limitations in non-discrimination law," AI and Ethics, **2**, 815–826, 2022, doi:10.1007/s43681-022-00142-y.

[31] M. Xue, Q. Huang, H. Zhang, L. Cheng, J. Song, M. Wu, M. Song, "Protopformer: Concentrating on prototypical parts in vision transformers for interpretable image recognition," arXiv preprint arXiv:2208.10431, 2022, doi:10.48550/arXiv.2208.10431.

[32] M. Leon Espinosa, G. Napoles Ruiz, "Modeling and Experimentation Framework for Fuzzy Cognitive Maps," Proceedings of the AAAI Conference on Artificial Intelligence, **30**(1), 2016, doi:10.1609/aaai.v30i1.9841.

[33] A. Adadi, M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)," IEEE Access, **6**, 52138–52160, 2018, doi:10.1109/ACCESS.2018.2870052.

[34] V. G. Costa, C. E. Pedreira, "Recent advances in decision trees: An updated survey," Artificial Intelligence Review, **56**(5), 4765–4800, 2023, doi:10.1007/s10462-022-10275-5.

[35] J. F. Allen, S. Schmidt, S. A. Gabriel, "Uncovering Strategies and Commitment Through Machine Learning System Introspection," SN Computer Science, **4**(4), 322, 2023, doi:10.1007/s42979-023-01747-8.

[36] A. Heuillet, F. Couthouis, N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," Knowledge-Based Systems, **214**, 106685, 2021, doi:10.48550/arXiv.2008.06693.

[37] P. Sequeira, E. Yeh, M. T. Gervasio, "Interestingness Elements for Explainable Reinforcement Learning through Introspection." in IUI workshops, volume 1, 2019, doi:10.48550/arXiv.1912.09007.

[38] X. Dai, M. T. Keane, L. Shalloo, E. Ruelle, R. M. Byrne, "Counterfactual explanations for prediction and diagnosis in XAI," in Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society, 215–226, 2022, doi:10.1145/3514094.3534144.

[39] M. T. Keane, E. M. Kenny, E. Delaney, B. Smyth, "If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques," arXiv preprint arXiv:2103.01035, 2021, doi:10.48550/arXiv.2103.01035.

[40] G. Warren, M. T. Keane, R. M. Byrne, "Features of Explainability: How users understand counterfactual and causal explanations for categorical and continuous features in XAI," arXiv preprint arXiv:2204.10152, 2022, doi:10.48550/arXiv.2204.10152.

[41] G. Nápoles, Y. Salgueiro, I. Grau, M. Leon, "Recurrence-Aware Long-Term Cognitive Network for Explainable Pattern Classification," IEEE Transactions on Cybernetics, **53**(10), 6083–6094, 2023, doi:10.48550/arXiv.2107.03423.

[42] P. A. Moreno-Sanchez, "An automated feature selection and classification pipeline to improve explainability of clinical prediction models," in 2021 IEEE 9th International Conference on Healthcare Informatics (ICHI), 527–534, IEEE, 2021, doi:10.1109/ICHI52183.2021.00100.

[43] J. Tritscher, A. Krause, A. Hotho, "Feature relevance XAI in anomaly detection: Reviewing approaches and challenges," Frontiers in Artificial Intelligence, **6**, 1099521, 2023, doi:10.3389/frai.2023.1099521.

[44] J. Tritscher, M. Ring, D. Schlr, L. Hettinger, A. Hotho, "Evaluation of post-hoc XAI approaches through synthetic tabular data," in Foundations of Intelligent Systems: 25th International Symposium, ISMIS 2020, Graz, Austria, September 23–25, 2020, Proceedings, 422–430, Springer, 2020, doi:10.1007/978-3-030-59491-6_40.

[45] G. Nápoles, F. Hoitsma, A. Knoben, A. Jastrzebska, M. Leon, "Prolog-based agnostic explanation module for structured pattern classification," Information Sciences, **622**, 1196–1227, 2023, doi:10.1016/j.ins.2022.12.012.

[46] Z. Yang, J. Liu, K. Wu, "Learning of Boosting Fuzzy Cognitive Maps Using a Real-coded Genetic Algorithm," in 2019 IEEE Congress on Evolutionary Computation (CEC), 966–973, 2019, doi:10.1109/CEC.2019.8789975.

[47] W. Liang, Y. Zhang, X. Liu, H. Yin, J. Wang, Y. Yang, "Towards improved multifactorial particle swarm optimization learning of fuzzy cognitive maps: A case study on air quality prediction," Applied Soft Computing, **130**, 109708, 2022, doi:10.1016/j.asoc.2022.109708.

[48] Y. Hu, Y. Guo, R. Fu, "A novel wind speed forecasting combined model using variational mode decomposition, sparse auto-encoder and optimized fuzzy cognitive mapping network," Energy, **278**, 127926, 2023, doi:10.1016/j.energy.2023.127926.

[49] W. Hoyos, J. Aguilar, M. Toro, "A clinical decision-support system for dengue based on fuzzy cognitive maps," Health Care Management Science, **25**(4), 666–681, 2022, doi:10.1007/s10729-022-09611-6.

[50] W. Hoyos, J. Aguilar, M. Toro, "PRV-FCM: An extension of fuzzy cognitive maps for prescriptive modeling," Expert Systems with Applications, **231**, 120729, 2023, doi:10.1016/j.eswa.2023.120729.

[51] G. Nápoles, I. Grau, L. Concepción, L. K. Koumeri, J. P. Papa, "Modeling implicit bias with fuzzy cognitive maps," Neurocomputing, **481**, 33–45, 2022.

[52] K. Poczeta, E. I. Papageorgiou, "Energy Use Forecasting with the Use of a Nested Structure Based on Fuzzy Cognitive Maps and Artificial Neural Networks," Energies, **15**(20), 7542, 2022, doi:10.3390/en15207542.

[53] G. D. Karatzinis, N. A. Apostolikas, Y. S. Boutalis, G. A. Papakostas, "Fuzzy Cognitive Networks in Diverse Applications Using Hybrid Representative Structures," International Journal of Fuzzy Systems, **25**(7), 2534–2554, 2023, doi:10.1007/s40815-023-01564-4.

[54] O. Orang, P. C. de Lima e Silva, F. G. Guimarães, "Time series forecasting using fuzzy cognitive maps: a survey," Artificial Intelligence Review, **56**, 7733–7794, 2023, doi:10.1007/s10462-022-10319-w.

[55] M. Leon, "Business Technology and Innovation Through Problem-Based Learning," in Canada International Conference on Education (CICE-2023) and World Congress on Education (WCE-2023), Infonomics Society, 2023, doi:10.20533/cice.2023.0034.

[56] E. Jiya, O. Georgina, A. O., "A Review of Fuzzy Cognitive Maps Extensions and Learning," Journal of Information Systems and Informatics, **5**(1), 300–323, 2023, doi:10.51519/journalisi.v5i1.447.

[57] R. Schuerkamp, P. J. Giabbanelli, "Extensions of Fuzzy Cognitive Maps: A Systematic Review," ACM Comput. Surv., **56**(2), 53:1–53:36, 2023, doi:10.1145/3610771.

[58] S. Yang, J. Liu, "Time-Series Forecasting Based on High-Order Fuzzy Cognitive Maps and Wavelet Transform," IEEE Transactions on Fuzzy Systems, **26**(6), 3391–3402, 2018, doi:10.1109/TFUZZ.2018.2831640.

[59] T. Koutsellis, G. Xexakis, K. Koasidis, N. Frilingou, A. Karamaneas, A. Nikas, H. Doukas, "In-Cognitive: A web-based Python application for fuzzy cognitive map design, simulation, and uncertainty analysis based on the Monte Carlo method," SoftwareX, **23**, 2023, doi:10.1016/j.softx.2023.101513.

[60] D. Qin, Z. Peng, L. Wu, "Deep attention fuzzy cognitive maps for interpretable multivariate time series prediction," Knowledge-Based Systems, **275**, 110700, 2023, doi:10.1016/j.knosys.2023.110700.

[61] G. P. Reddy, Y. P. Kumar, "Explainable AI (XAI): Explained," in 2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 1–6, IEEE, 2023, doi:10.1109/eStream.2023.00001.

[62] Y. Zhang, Y. Weng, J. Lund, "Applications of explainable artificial intelligence in diagnosis and surgery," Diagnostics, **12**(2), 237, 2022, doi:10.3390/diagnostics12020237.

[63] A. Nielsen, S. Skylaki, M. Norkute, A. Stremitzer, "Effects of XAI on Legal Process," ICAIL '23: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, 2023, doi:10.1145/3593013.3594067.

[64] P. Weber, K. V. Carl, O. Hinz, "Applications of Explainable Artificial Intelligence in Finance—a systematic review of Finance, Information Systems, and Computer Science literature," Management Review Quarterly, **74**, 867–907, 2023, doi:10.1007/s11301-023-00320-0.

[65] H. DeSimone, M. Leon, "Explainable AI: The Quest for Transparency in Business and Beyond," in 2024 7th International Conference on Information and Computer Technologies (ICICT), 1–6, IEEE, 2024, doi:10.1109/icict62343.2024.00093.

[66] G. Nápoles, J. L. Salmeron, W. Froelich, R. Falcon, M. Leon, F. Vanhoenshoven, R. Bello, K. Vanhoof, Fuzzy Cognitive Modeling: Theoretical and Practical Considerations, 77–87, Springer Singapore, 2019, doi:10.1007/978-981-13-8311-3_7.

[67] L. Longo, R. Goebel, F. Lecue, P. Kieseberg, A. Holzinger, "Explainable artificial intelligence: Concepts, applications, research challenges and visions," in International Cross-Domain Conference for Machine Learning and Knowledge Extraction, 1–16, Springer, 2020, doi:10.1007/978-3-030-57321-8_1.

[68] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, "Explainable AI: A brief survey on history, research areas, approaches and challenges," in Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II, 563–574, Springer, 2019, doi:10.1007/978-3-030-32236-6_51.

[69] W. Saeed, C. Omlin, "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities," Knowledge-Based Systems, **263**, 110273, 2023, doi:10.48550/arXiv.2111.06420.

[70] A. Toosi, A. G. Bottino, B. Saboury, E. Siegel, A. Rahmim, "A brief history of AI: how to prevent another winter (a critical review)," PET Clinics, **16**(4), 449–469, 2021, doi:10.1016/j.cpet.2021.07.001.

[71] T. Hulsen, "Explainable Artificial Intelligence (XAI): Concepts and Challenges in Healthcare," AI, **4**(3), 652–666, 2023, doi:10.3390/ai4030034.

[72] R. Justo-Hanani, "The politics of Artificial Intelligence regulation and governance reform in the European Union," Policy Sciences, **55**(1), 137–159, 2022, doi:10.1007/s11077-022-09452-8.

[73] P. Hacker, A. Engel, M. Mauer, "Regulating ChatGPT and other large generative AI models," in Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 1112–1123, 2023, doi:10.1145/3593013.3594067.

[74] E. Parliament, "EU AI Act: first regulation on artificial intelligence," 2023.

[75] C. Panigutti, R. Hamon, I. Hupont, D. Fernandez Llorca, D. Fano Yela, H. Junklewitz, S. Scalzo, G. Mazzini, I. Sanchez, J. Soler Garrido, et al., "The role of explainable AI in the context of the AI Act," in Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 1139–1150, 2023, doi:10.1145/3593013.3594069.

Advances in Science, Technology and Engineering Systems Journal Vol. 9, No. 5, 21-29 (2024)

www.astesj.com

Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ

ISSN: 2415-6698

Evaluation of a Classroom Support System for Programming Education Using Tangible Materials

Koji Oda* [1], Toshiyasu Kato [2], Yasushi Kambayashi [3]

[1]*Department of Information and Telecommunication, Saitama Prefectural Kawaguchi Technical High School, Kawaguchi, 333-0846, Japan*

[2]*Department of Information and Media Engineering, Nippon Institute of Technology, Minamisaitama, 345-0826, Japan*

[3]*Department of Informatics and Data Science, Sanyo-Onoda City University, Sanyo-Onoda, 756-0884, Japan*

ARTICLE INFO

Article history:
Received: 14 September, 2024
Revised: 08 October, 2024
Accepted:09 October, 2024
Online: 18 October, 2024

Keywords:
Tangible materials
Programming education
Classroom support systems
Face-to-face instruction

ABSTRACT

In recent years, the utilization of tangible educational materials has attracted attention on educational settings. They provide hands-on learning experiences for beginners. This trend is especially notable in the field of programming education. Such educational materials are employed in many institutions worldwide. They liberate learners of programming from programming languages that are confined in a small computer screen. On the other hand, in the school setting, classroom time is limited. When instructing more than thirty students, it is hard for instructors to provide adequate guidance for everyone. To address this problem, we have developed a classroom support system for programming education that complements the use of tangible educational materials. With this system, instructors can monitor the real-time progress of each student during the class and analyze which parts of the program many students find challenging. Based on these analytical results, instructors can provide appropriate instructions for individual students and effectively conduct the class. This system is suitable for programming education in high schools. It quantifies each student's ability of programming and track the progress of each student. We administered a questionnaire to both the students and the instructor. The results of the questionnaire show our system is well received by both students and the instructor. Even though our system demonstrates some usefulness for programming beginners, we are aware that our system has some serious limitations such as our rigid model answers.

1. Introduction

This paper is an extension of work originally presented in *2024 Twelfth International Conference on Information and Education Technology* (ICIET 2024) [1]. The work presented the basic idea of system and the results of the preliminary experiments that indicated its usefulness. In this paper, we have extended the paper to explain our system in details and to demonstrate its effectiveness through showing results of larger scale experiments. For programming beginners, numerous GUI programming systems have been proposed. However, the computer screen and the display resolution restrict the students' recognizability of program elements. This problem makes the programming activities difficult especially with lower resolution displays. To address this issue, we developed tangible educational materials named "Jigsaw Coder" for programming education [2]. In the following, we will refer to this as JC. JC consists of multiple cards. Each card has QR code printed on it, and students can construct programs by rearranging them. This enables programming on a desk or even on the floor, which provides much larger space. The user can take a photo to read the complete program by their smartphones and also execute the program on their smartphone. However, such tangible educational materials were designed for self-taught of individual learners. It is challenging for class room use; it is hard for instructors to grasp the progresses of all students when used in a class of more than a few, e.g. thirty, students. The objective of this study is to design and to implement a system that provides instructors information of real-time progresses of the students so

*Corresponding Author: Koji Oda, Saitama Prefectural Kawaguchi Technical High School. mooda194lun@yahoo.co.jp

that he or she can analyses information of students' programming in classes using JC. The system helps instructors to practice much effective use of instruction time.

The authors conducted a preliminary evaluation of JC as prior research [1]. As a result of performing a functional check assuming an actual class, there were no issues with the system's operation with around ten users, and it was possible to conduct a trial evaluation simulating an actual class. This paper demonstrates the effectiveness of JC through an evaluation experiment conducted in actual high school classes.

2. Research Methodology

We have developed a tangible programming system that utilizes JC and Micro:bit for educational purpose. This system allows students to engage in tangible programming, while instructor can monitor their progresses in real-time and perform analysis over their achievements. Subsequently, we conducted classes as part of the evaluation experiments and administrated questionnaires for both instructor and students to assess the effectiveness of the system. In the previous papers, we reported our development and evaluation of the tangible educational materials [3, 4]. The materials involve rearranging multiple cards to program. Then the user makes the system read the QR codes printed on them using a smartphone to execute the program. We call this card-type tangible educational system as JC. Figure 1 shows the flow of the programming process. In the original JC, we used a smartphone; in this study, we decided to utilize Chromebooks, because they are easy to use and widely adopted in many Japanese schools.

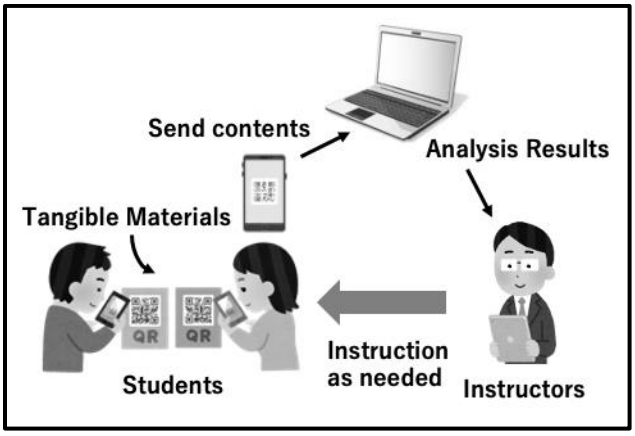


Figure 1: Instruction utilizing tangible education materials

2.1. JC

A client PC (Chromebook) creates a program from QR cards and writes it to the Micro:bit. Simultaneously, the program code is transferred to the server. The server then analyzes the received program code. The analysis flow is as follows.

The server saves the program code as a file and compares it with the corresponding model answer. In the comparison process, it calculates the matching rate with the model answer and identifies the positions of incorrect sections. The answer data for each student--such as student name, first answer time, most recent answer time, final answer time, number of responses, matching rate with the model answer, line numbers and positions of mistakes, and program level--is stored in the database. Subsequently, a web page reads the database and displays the answer information for each student. At this point, based on the answer information, students are classified into three categories: Unanswered, Progress, and Completed. This allows the instructor to easily track each students' progress at a glance. Additionally, a page is generated that allows the instructor to review each students' answer. On this page, it is easy to identify missing, extra, or incorrect parts of the answer. Based on this information, the instructor can provide specific feedback to the students.

2.2. Micro:bit

Micro:bit is a microcontroller designed by the British Broadcasting Corporation (BBC) for programming education. It can display characters and shapes on LEDs and produce sound through a speaker. It also features sensors such as an accelerometer, magnetometer, microphone, temperature sensor, and light sensor, which allow it to recognize vibrations and changes in its environment. Additionally, Micro:bit includes wireless communication capabilities, enabling it to communicate with other Micro:bit. Programming can be done via a browser or app, and programs can be transferred to the Micro:bit for execution. Figure 2 shows a Micro:bit.

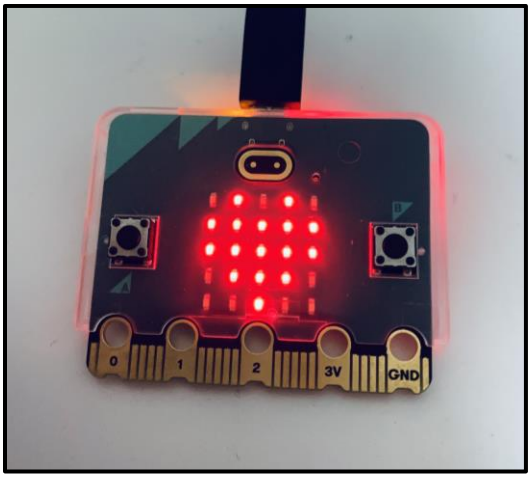


Figure 2: Micro:bit

3. Design And Implementation

We developed our system using Python. The reason for choosing Python is its high readability due to a vast array of libraries. This fact let us build shorter programs. In addition, Python is an interpreted language, enabling immediate execution without compilation, making it suitable for creating prototypes. To run the proposed system, some preparations are needed. The preparation before the class includes:

1) Creating tasks for students (assigning unique task numbers).
2) Creating and placing example answer programs and level configuration files.
3) Inputting students' information.

Carrying out a programming class includes:

1) Starting the server and server program.

2) Connecting Micro:bit to student's Chromebook.

3) Starting the client program.

3.1. Improvement of Jigsaw Coder

In this project, we added three more elements to enable more intuitive rearrangements. The first element is emphasizing the task number. To distinguish which task the student is working on, he or she initially needs to make the system read the QR code for the task number card in JC. Then, the background color of the task number card was changed, and highlighted the numbers by surrounding them with star symbols. The second element is the use of symbols "⊳" and "⊲". These symbols represent the role of "{" and "}" in the conventional programming languages such as C and Java. They are used to denote looping constructs like "Repeat ⊳" and "⊲ End here," aiding in the intuitive understanding of grouping. The third element is "→" and "←", representing arranging cards side by side. These symbols are utilized when specifying conditional statements, such as "If Condition →" and "← Press A Button ⊳". These symbols help learners intuitively grasp the utilization and representation of conditions. Figure 3 shows the cards used by the students.

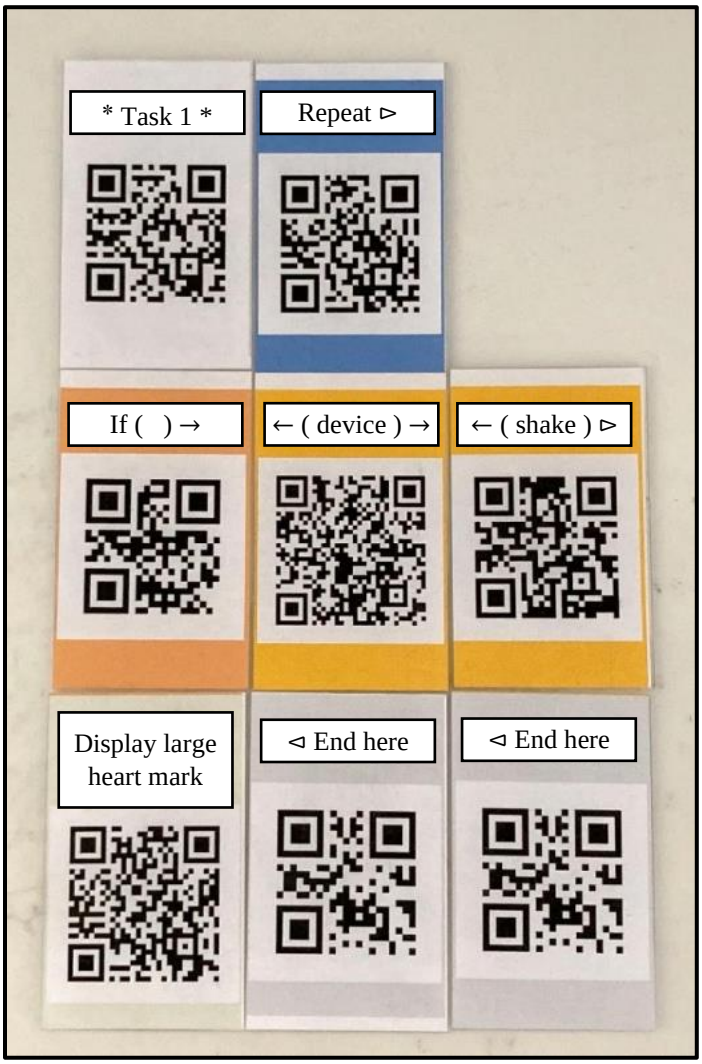


Figure 3: QR cards used in JC

3.2. Operation of the Students' Side (Client Program)

Figure 4 shows the flow of operations for the client program.

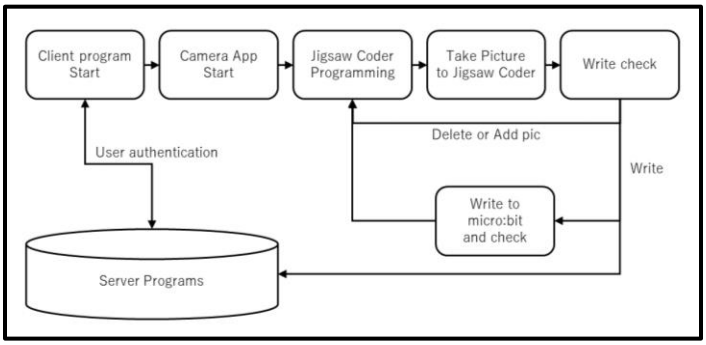


Figure 4: Flow of client program

Upon starting the client program, student authentication is initiated. The system prompts the student to input the grade, class, and the student number. Upon pressing the confirm button, the connection with the server program is established, and the students' name is displayed. Figure 5 shows the user authentication window.

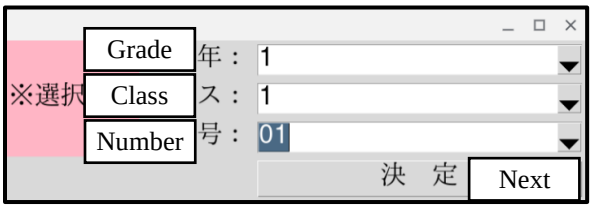


Figure 5: User authentication window

After completing student authentication, students begin programming. Once the students complete their arrangements of the cards, they photograph the cards using the camera application. The client program reads the captured photo, analyzes the QR codes in order, and generates the corresponding program. The captured photos are deleted to save memory space as they are no longer needed. Students can review the generated program in a window and then write it to the Micro:bit after confirmation. Figure 6 shows the confirmation window.

Figure 6: Writing confirmation window

If "Read additional" is selected, the read program is temporarily saved, and the student can capture another photo as the continuation of the program using the camera application. If "No (initialize content)" is selected, the read program is deleted, and the students can take a new photo again from the beginning. If "Yes" is selected, the system initiates the writing process to the

Micro:bit connected to the Chromebook. At this point, the student sends the program they wrote to the server program. The student checks their Micro:bit to ensure that the program is running correctly. If errors are found, the student rearranges the cards and takes another photo again. Figure 7 displays a photo of the system used by students.

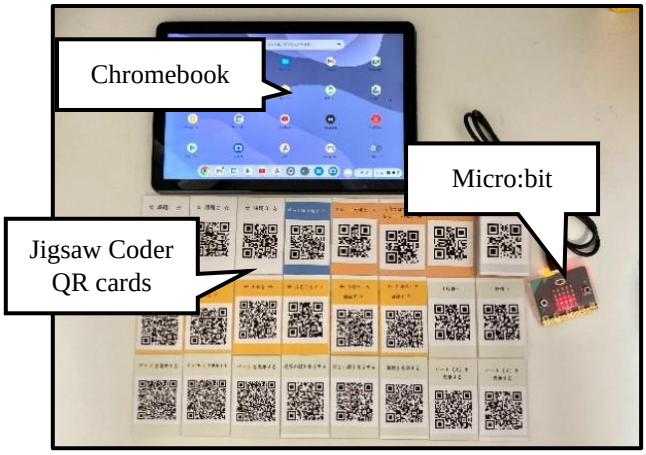


Figure 7: Overview of the student side system of JC

3.3. Operation of the Instructor's Side (Server Program)

Within the server, a database is set to manage a list of students and multiple tasks for them. The database contains a table for the student list, where their information is pre-stored for student authentication purposes. On the other hand, the task table maintains student progress, including the date and time the student first answered, date and time of subsequent attempts, date and time of correct answers, number of attempts, position of incorrect parts of their programs compared to the example answer program and level configuration file, match rate, and the program level calculated from the level configuration file.

3.4. The Web Page for Analysis

The instructor reviews the information in the database on a web page using a browser. This web page accesses the database using PHP and presents the information in an easy-to-review format for the instructor. Figure 8 shows the top page, where the number of programs in progress and completed answer programs for each task are summarized in a table. From the student list page, the instructor can edit or delete student information. It is also possible to import student data from an Excel file.

Each task page consists of two segments. Figures 9 and Figure 10 show the pages for a task. At first, using the first segment, instructor analyzes the parts of the program where students frequently make mistakes. The segment displays an example answer program, highlighting the background color of the areas where many students make mistakes. The background color changes from blue to yellow to red as the number of students who make mistakes increases in that particular section. The second segment is the table summarizing the progress of each student. It is divided into three tables for not answered, in progress, and completed of the programs, compiling details such as date and time of answer, number of attempts, and positions of errors. Using this table, the instructor can grasp the progress of the entire class. Additionally, by selecting a students' ID in this table, the instructor can see a page that compares the students' program and the example answer program for that specific student. Figure 11 shows the comparison page.

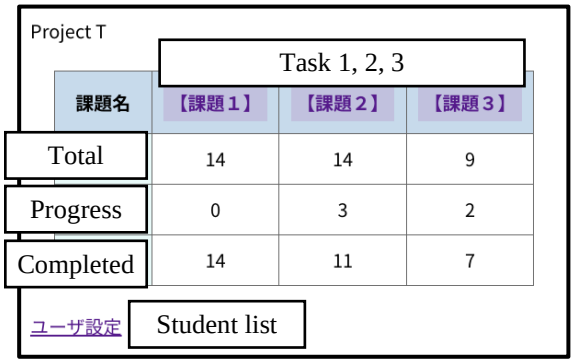


Figure 8: Analysis table on the homepage

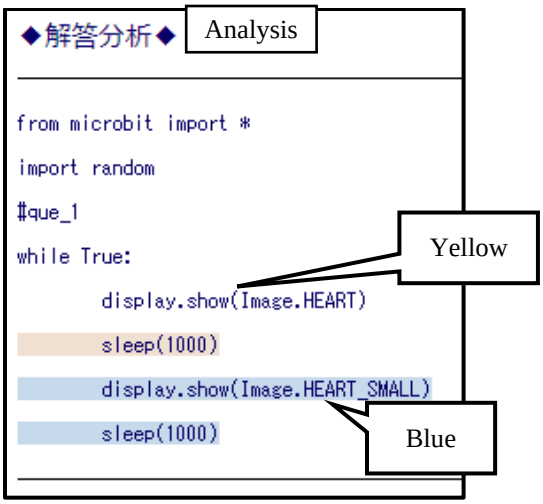


Figure 9: Segment for analyzing answer

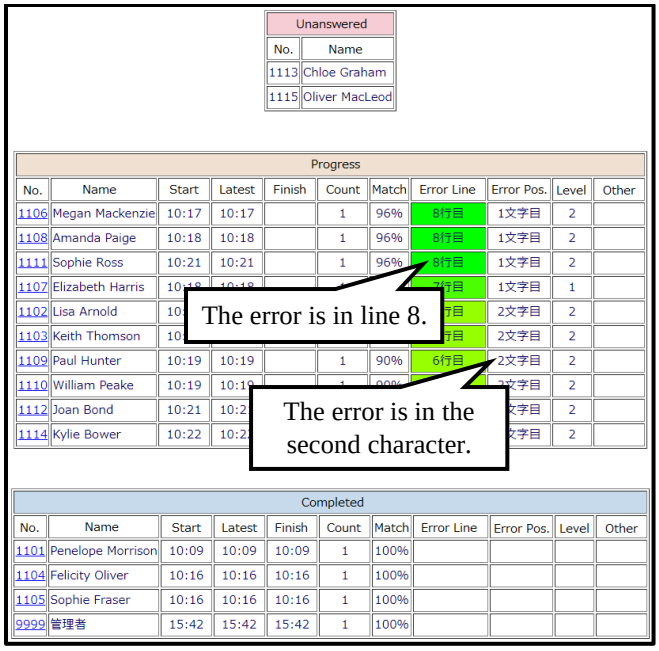


Figure 10: Segment for display answers list

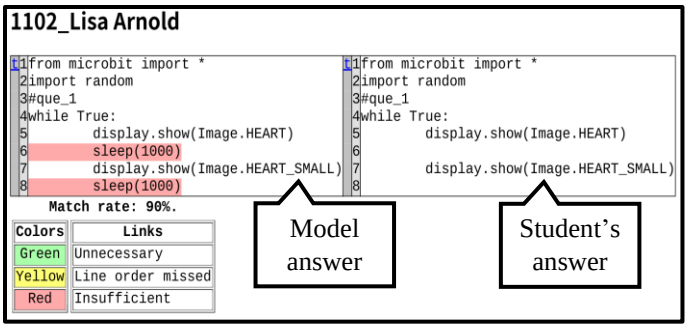


Figure 11: Page for comparing model answer with students' answer

4. Evaluation Experiment

In order to demonstrate the effectiveness of our system, we conducted evaluation experiments on the system. The objectives of the experiments are as follows.

- To determine whether the system can function properly for a large number of students during actual class time.
- If any delays occur, to measure the duration of these delays.
- From the students' perspective, we evaluated "Overall system feedback," "Feedback on sequential processing, repetition, and branching in programming," and asked "Whether programming beginners can develop an interest in programming."
- From the instructor's perspective, we evaluated "Overall system feedback," "Convenience of monitoring student progress," and asked "Areas of potential improvement in the system."

To confirm the above objectives, we let a high school instructor conduct an actual programming class. Afterwards, we administered questionnaires to both the students and the instructor. The students are nineteen first-year students from Gunma Prefectural Annaka General Academic High School. The students had no prior programming experience. The tasks prepared for this evaluation experiment were as follows:

1) Display a large heart mark and a small heart alternately for 1 second each.
2) Pressing the "A" button when the device displays a smiling face, pressing the "B" button when it displays a sad face, and not pressing any button when it displays a neutral face.
3) Shaking the device displays either rock, paper, or scissors for a rock-paper-scissors game.

The objective of task 1 is to facilitate learning of sequential processing and repetition. Task 2 aims to utilize button inputs and learn about branching. Task 3 is an optional task. The goal of this task is to utilize shaking the device as an input and understand the multi-level branching in the context of learning. All tasks involve elements of repetition.

4.1. Experimental Results

We found the system is stable. The instructor felt no perceivable delays. However, on the client side, there were instances where the program stops due to students' input errors.

4.2. Results of the Student Survey

The followings are the questions for the student survey:

1) Did using this material help you grasp the basic structure of a program?
2) Did experiencing this material increase your interest in programming?
3) Please indicate your perceived level of understanding of sequential processing.
4) Please indicate your perceived level of understanding of iterative (repetitive) processing.
5) Please indicate your perceived level of understanding of conditional processing.
6) Please write what you felt on the materials and lessons used in this session.

The questions and answers from the student survey correspond to Figures 12 through 16.

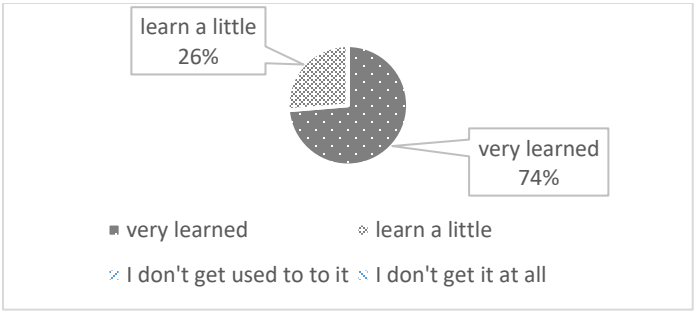


Figure 12: Did using this material help you grasp the basic structure of programming?

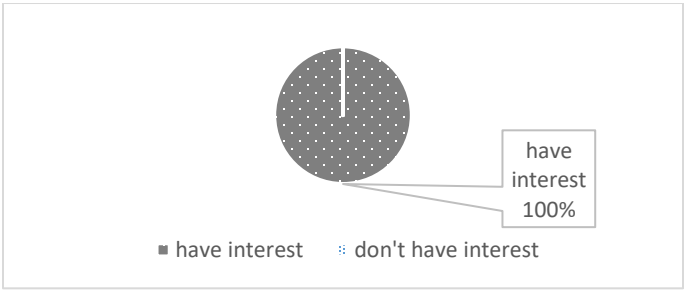


Figure 13: Did experiencing this material increase your interest in programming?

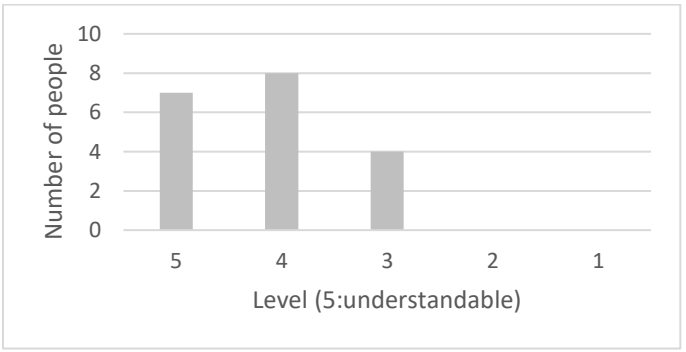


Figure 14: Please indicate your perceived level of understanding of sequential processing.

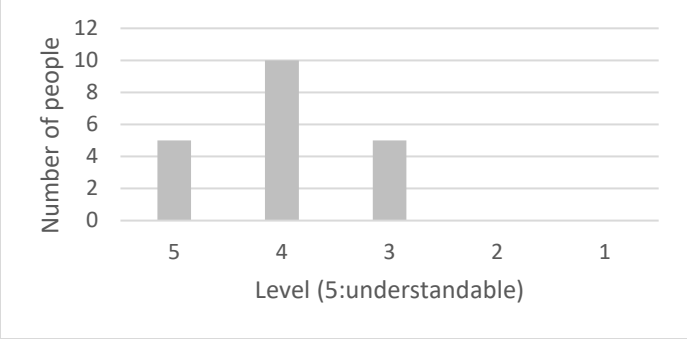


Figure 15: Please indicate your perceived level of understanding of iterative (repetitive) processing.

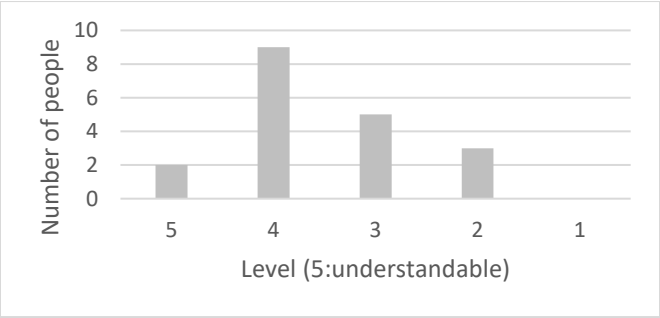


Figure 16: Please indicate your perceived level of understanding of conditional processing.

As an answer to question 6 (Please write what you felt on the materials used in this class.), the following feedbacks were provided:

- The steps were easy to remember.
- The cards were heavy.
- Having the materials allowed for better communication between students and instructors.
- I was able to think and work on my own.
- I couldn't solve the conditional processing problems.

4.3. Results of the Instructor Survey

After class, we conducted a survey for the instructor. The questions and answers are as follows. Questions A through C use a 4-point scale, where 1 indicates the lowest rating and 4 indicates the highest rating, respectively. Table 1 shows the results.

A) Did you comprehend the overall class situations based on the presented analysis results?

B) Did you discover specific problems based on the presented analysis results?

C) Did the classification based on learning levels from the presented analysis results assist for the instruction?

Table 1: Results of the survey for instructor

Question	Answer
A	2
B	3
C	1

D) Please tell us what you like about this system.

- The tangible materials, involving the combination of physical cards, are promising as an introductory tool for those new to programming.
- Taking photos of the program cards is easy and accurate enough.
- We can monitor students' progresses without moving around the classroom.
- We can focus on the students with many errors.
- The three tasks within a two-hour class is appropriate.
- Instead of using the system for real-time monitoring during class, it might be beneficial as a self-learning tool. Results, including errors, could inform instruction for future classes.

E) Please tell us any dissatisfactions or points for improvement of this system.

- It is difficult to take pictures because of the wired Micro:bit connection.
- There were many connection errors with the Micro:bit. I It needs to be improved. It was hard to tell whether it is a connection error or a programming error. It would be good to have an indication lamp or beep sound for that.
- I is unable to identify which part of the program students are struggling.
- When instructors inspect students' programs, they see the corresponding Python code instead of JC cards. It is stressful for instructors without sufficient programming skills.

5. Discussion

This section analyzes and discusses the results of the evaluation experiments.

5.1. Discussion Based on the Student Survey

Survey results indicate positive feedback on the materials. Question 2 reveals that our system successfully developed intellectual inquisitiveness for programming in all the students. Since all the students were beginners, the system effectively achieved its goal of generating motivation for programming. For question 3 on sequential processing, there were many positive responses. Students understood the order of operations by rearranging the cards. This suggests that the card arrangement helped clarify sequential processing. Question 4 on iterative processing also received positive feedbacks. In contrast, question 5 on conditional processing had a lower average rating of 3.53. This lower rating may be due to the task's difficulty. Task 3, designed to teach conditional processing, required two conditionals, which might be challenging. Starting with simpler tasks could improve understanding of conditional processing. Additionally, students might have struggled with the visual and intuitive differences between "if" and "else if," as well as between "→" and "⊳." We need to reconsider the design of JC to make these concepts more intuitive and easier to grasp. We plan to have different notations in the next version.

5.2. Discussion Based on the Instructor Survey

We can obtain several insights from the instructor survey. Questions A and C received negative responses. They indicate problems with the current system. Although instructors could track students' progress without moving around the classroom, they struggled to identify overall student difficulties. Positive responses to question B show the system is effective in identifying issues of individual students. However, question E responses indicate that how the instructor feel the system depends on their background knowledge of Python. The system requires instructors to read Python code on their screens, which may be problematic if their programming skills are insufficient. We may need to reconsider our assumption that the instructor should have sufficient programming skill in Python, and how to show students' progresses to the instructors. We plan to forge a novel means to display students' progresses so that it enhances the system's accessibility and effectiveness for users with varying levels of programming expertise.

5.3. Discussion of the Overall System

The system has not faced performance issues like delays so far. However, future experiments with over thirty students may present challenges. Unforeseen issues such as delays could arise depending on the server's capacity. Currently, a LAMP environment on a Chromebook is used for testing, but a dedicated server may be needed for practical use. Processing programs also needs adjustment to accommodate more number of users. Ensuring the system remains functional even when users cause serious run-time errors is crucial. For instance, adding confirmation dialogs to prevent accidental stops of the client program could reduce such opportunities. The card recognition issues, such as when only nine out of ten cards are recognized due to environmental factors, suggest the need for improving such as providing a new confirmation window.

We are implementing such a confirmation window. Figure 17 shows the new confirmation window that replaces the one shown in Figure 6. Since the message displayed in this window is written in Japanese, we show the corresponding English translation in Figure 18. Displaying the text on the cards before transferring to the Micro:bit could ensure correct card recognition. This approach helps students review their work and strengthen their understanding of both tangible and text-based programming.

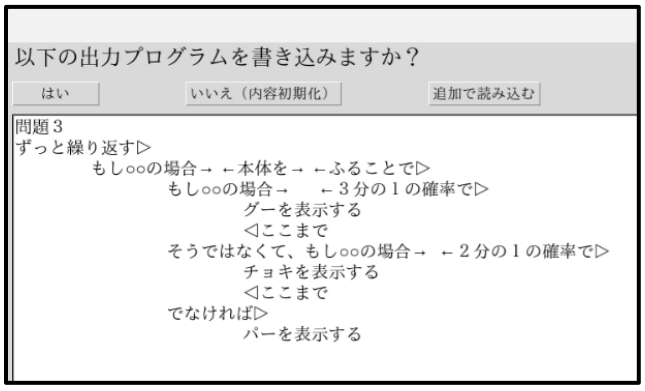


Figure 17: Writing confirmation window in Japanese

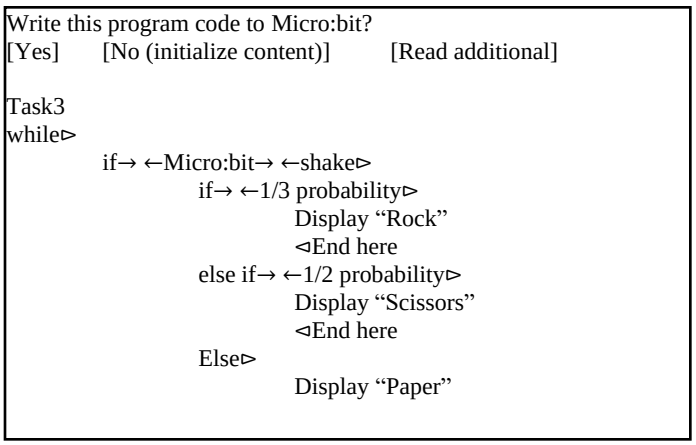


Figure 18: Writing confirmation window in English

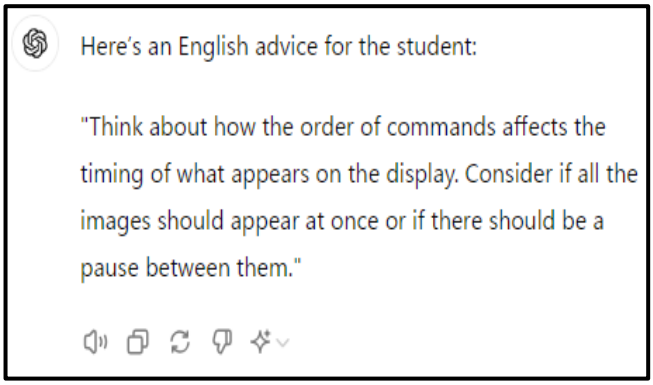


Figure 19: An example of advice generation using ChatGPT

The current limitation of this system is that only one model answer can be set for each task. Since current problem set includes only simple problems, one model answer for each problem is sufficient. We plan to incorporate a parser in our system, and to utilize AI to interpret responses more flexibly. This could identify not only syntax errors, but also semantic errors and runtime errors and provide tailored feedback for each program. Figure 19 shows an example of generated advice. While this example uses a GUI-based ChatGPT, we plan to incorporate the Python API for faster processing.

We are planning to create an individual page for each student. These pages would show the students' progress and provide AI-generated advice on each program so that each students can access to the information tailored for each of them and learn at any time. Students' feedback on the JC materials reveals that the cards are heavy. Currently, acrylic boards are used, which are durable for younger students. We need to explore alternative materials for the cards. Additionally, we are also considering modify the shapes of the cards related to "if," "else if," "else," and "while." This change aims to make the concepts of branching and iteration more intuitive.

6. Related Works

We referred to the literature on the development of tangible educational materials, literature on group learning analysis, and literature on education using Micro:bit, as listed below. Many related studies aim at the development of programming

Table 2: Comparison with other tangible educational materials

Name	Tangible	Analysis Multi-Student	Guidance for Struggling Students	Analysis on Class	Analysis after Class	Generating Individual Feedback
Jigsaw Coder	+	+	+	+	+	-
T-Maze [5]	+	-	-	-	-	-
Plugramming [6]	+	-	-	-	-	-
Strawbies [7]	+	-	-	-	-	-
PaPL [8]	+	+	-	-	+	-
Kamichi's System [9]	-	+	-	+	+	+
Kato's System [10]	-	+	+	+	+	-

educational materials. Wang et al. developed and evaluated a programming-based maze escape game called "T-Maze" [5]. However, environments with multiple students like those in a school classroom setting were not taken into consideration. Tomohito Yashiro et al. developed a tool called "Plugramming" and conducted the construction and evaluation of a collaborative programming system using Scratch [6]. However, it fails to address situations where multiple students stumble at similar parts and encounter similar errors. Felix Hu et al. developed a tangible programming game called "Strawbies" for children aged 5 to 10 [7]. Programming is done using wooden tiles. Since the tiles are not square but have distinctive shapes, the users cannot make incorrect connections. Although the programming flexibility is reduced, it has the advantage of intuitively understandable whether a connection is possible or not. Aditya Mehrotra et al. proposed multiple approaches for programming education conducted in a classroom setting [8]. They utilized program blocks for robot programming and evaluated several methods. However, the evaluation was aimed at assessing the methods, and the system does not provide real-time instructions based on students' progresses. It does not promote knowledge retention either.

In many studies related to programming instruction, the main objective is to support programming. Koichi Kamichi designed a system for programming education without teaching assistants [9]. The system mirror leaners answer to the server, providing automated suggestions of input errors for students and allowing monitoring of the progresses of the students. However, the automatic suggestions for input errors primarily aims to detect syntax errors, not considering programming novices who lack knowledge about logical thinking, which are prerequisites. Furthermore, the system only provides the instructor the number of errors that the student made, and does not provide more detailed analyses. Kato et al. developed a system in which they collected and analyzed the progress of students' programming in classes with teaching assistants and utilized this information effectively for teaching assistants so that they can guide students efficiently [10]. They conducted evaluation experiments demonstrating the system's effectiveness in instructional support. However, this analysis focuses on traditional programming languages and cannot be applied to tangible teaching materials.

Michail et al. systematically reviewed and summarized how the Micro: bit is used in primary education [11]. They reported that many students enjoy to use Micro:bit and found it easy to use. They evaluated it as beneficial for improving programming skills. In the survey, they demonstrated that Micro:bit is a promising tool for approaching STEM education. Dylan et al. conducted a two-week Micro:bit programming education program with 41 high school students[12]. After experiencing basic Micro:bit programming, the students became to be able to program autonomous cars equipped with Micro:bit and ultrasound distance sensors. Pre- and post-tests were conducted, and the results showed that the students' understanding of information processing and algorithms had deepened.

7. Conclusion

In this study, we reported our experiences of development of a classroom support system. This system assists students in programming and instructors who teach them. We conducted evaluation experiments to demonstrate the effectiveness of our system. We show a comparison of Jigsaw Coder (JC) with other related systems in Table 2. In general, it is difficult to monitor each student's progress in programming classes, and JC solves this issue. JC collects and analyzes each student's answer, and provides the instructor information for effective instruction. JC points out program areas where many students are making mistakes in the class. This function helps the instructor to grasp the overall status of the class without inspecting students one by one. Furthermore, JC is a tangible learning system, and it allows students to learn programming through physical interaction. As long as a school can provides Micro:bits, paper QR cards, client PCs, a server PC, and a network, JC can be used in all economic regions around the world. Especially it is beneficial for students in developing countries. We conducted evaluation experiments of JC for high school students. They are new to programming. The students' responses were generally positive. The instructor's responses were also positive that JC could serve as an entry-level tool for programming. It allows the instructor to monitor the students' progress without moving around the classroom to check students one by one. Based on these results, we believe that JC is effective for programming education at a beginners level. On the other hand, authors are aware that the system has a serious limitation. We plan

to revise the system with a parser and an analyzer to assist students building programming skills as well as logical thinking abilities. We reconsider the QR card design and try to make it simple too.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number JP21K02805.

References

[1] K. Oda, T. Kato, Y. Kambayashi, "Development and Evaluation Experiment of a Classroom Support System for Programming Education Using Tangibles Educational Materials," Proceedings of the 12th International Conference on Information and Education Technology (ICIET), 67-71, 2024, DOI: 10.1109/ICIET60671.2024.10542715

[2] T. Kato, K. Oda, Y. Kambayashi, "A Proposal of Educational Programming Environment Using Tangible Materials," Human Systems Engineering and Design (IHSED2023), 1-8, 2023.

[3] Y. Kambayashi, K. Furukawa, M. Takimoto, "Design of Tangible Programming Environment for Smartphones," HCI 2017: HCI International 2017, 448-453, 2017, DOI: 10.1007/978-3-319-58753-0_64

[4] Y. Kambayashi, K. Tsukada, M. Takimoto, "Providing Recursive Functions to the Tangible Programming Environment for Smartphones," HCII 2019, 255-260, 2019, DOI: 10.1007/978-3-030-23525-3_33

[5] D. Wang, C. Zhang, H. Wang, "T-Maze: A Tangible Programming Tool for Children," IDC '11: Proceedings of the 10th International Conference on Interaction Design and Children: 127-135, 2011, DOI: 10.1145/1999030.1999045

[6] T. Yashiro, K. Mukaiyama, Y. Harada, "Programming Tool and Activities for Experiencing Collaborative Design," Information Processing Society of Japan, **59(3)**: 822-833, 2018.

[7] F. Hu, A. Zekelman, M. Horn, F. Judd, "Strawbies: Explorations in Tangible Programming," IDC '15: Proceedings of the 14th International Conference on Interaction Design and Children: 410-413, 2015, DOI: 10.1145/2771839.2771866

[8] A. Mehrotra, C. Giang, N. Duruz, J. Dedelley, A. Mussati, M. Skweres, F. Mondada, "Introducing a Paper-Based Programming Language for Computing Education in Classrooms," ITiCSE '20: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education: 180-186, 2020, DOI: 10.1145/3341525.3387402

[9] K. Kamichi, "Designing a Programming Education Support System for Lessons without Practical Assistants," Journal of Sociology Research Institute, **1**: 73-78, 2020.

[10] T. Kato, Y. Kambayashi, Y. Kodama, "Data Mining of Students' Behaviors in Programming Exercises," Smart Education and e-Learning, **59**: 121-133, 2016.

[11] M. Kalogiannakis, E. Tzagaraki, S. Papadakis, "A Systematic Review of the Use of BBC Micro in Primary School," 10th International Conference New Perspectives in Science Education, STEM5036, 2021.

[12] D. G. Kelly, P. Seeling, "Introducing Underrepresented High School Students to Software Engineering: Using the Micro Microcontroller to Program Connected Autonomous Cars," Computer Applications in Engineering Education, **28(3)**: 737-747, 2020, DOI: 10.1002/cae.22244

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 9, No. 5, 30-36 (2024)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

True Random Number Generator Implemented in ReRAM Crossbar Based on Static Stochasticity of ReRAMs

Tanay Patni*, Abhijit Pethe

Department of Electrical and Electronics Engineering,BITS Pilani K.K. Birla Goa Campus, Goa, India, 403726, India

ARTICLE INFO

Article history:
Received: 31 July, 2024
Revised: 05 October, 2024
Accepted: 06 October, 2024
Online: 30 November, 2024

Keywords:
TRNG
Memristors
ReRAM Crossbar
Static stochasticity

ABSTRACT

True Random Number Generators (TRNG) find applications in various fields, especially hardware security. We suggest a TRNG that exploits the intrinsic static stochasticity of Resistive Switching Random Access Memories (ReRAMs) to generate random bits. Other suggested designs use stochasticity in the switching mechanism, which requires high precision over input voltage and time. In the proposed design, the random bits are produced by comparing the resistance of two ReRAMs in their high resistance states. ReRAM crossbar architectures are being highly researched, and our design is completely compatible with a ReRAM crossbar. The design was verified by simulations and testing the output stream using the NIST randomness test suite. The effect of device-to-device variability was tested on the randomness of the generated output bit stream.

1. Introduction

This paper is an extension of work originally presented in The IEEE Asia Pacific Conference On Circuits And Systems (APCCAS 2023) [1]. Random Numbers find a lot of applications in various fields, including scientific simulations and modeling, games, machine learning, and, most importantly, generating cryptographic keys [2, 3, 4]. Random numbers are generated using specialized hardware called Random Number Generators (RNGs). There are two types of RNGs, Pseudo Random Number Generators (PRNGs) and True Random Number Generators (TRNGs), differentiated based on the principle of number generation. PRNGs generate random numbers using algorithms based on mathematical formulae. While PRNGs are suitable for other applications, they cannot be used for security applications as they are vulnerable to attacks [5, 6], compromising security. TRNGs exploit the stochasticity of physical processes, e.g., Thermal Noise in electrical circuits [7], to generate random numbers. Since the source of randomness in TRNGs is inherently stochastic, they, in principle, can guarantee absolute information security.

Recently, there has been an increase in IoT devices in the market, which are small and have a small power budget. Since they continuously transmit confidential and private information, there is a need for a robust security system within the devices, necessitating a suitable TRNG to generate random numbers for encryption [8, 9]. Current TRNG circuits are made of transistors and are based on thermal noise, jitter in oscillators, random telegraph noise, or chaotic systems [10, 11, 12]. These circuits are bulky, complicated, and consume a lot of power, making them unsuitable for IoT devices.

ReRAM devices can be used as an alternative to design TRNGs. ReRAMs are emerging non-volatile memory devices extensively researched for crossbar architecture. This crossbar architecture finds applications in non-volatile logic, neuromorphic computing, security, in-memory computing, etc. [13, 14]. They consume low power, are small, are compatible with the CMOS fabrication process, and have fast switching speeds. They are also inherently stochastic, making them a good alternative for TRNG circuit design. ReRAMs exhibit stochasticity in two ways – during switching and the resistance values of the stable states. Many ReRAM-based TRNG designs have been suggested in the literature before, mainly focusing on switching stochasticity [15, 16, 17]. These designs require very precise control over voltages and timing, making the circuits complicated to implement. The variability in the resistance value can also be exploited to design TRNG circuits. Since they do not require precise control of input signals, they are easier to implement. One such design compares the resistance value of two devices to extract the output bit [18].

We propose a TRNG circuit based on the above principle, implementable in a ReRAM crossbar. This enables in-situ random number generation for crossbar applications and eliminates the need for a specialized TRNG circuit. The proposed circuit is simulated in Cadence Virtuoso™, and the randomness of the output is verified using the NIST SP 800-22 test suite [19]. We further analyzed the effect of variation in the statistical properties of ReRAM stochas-

*Corresponding Author: Tanay Patni, f20201745@goa.bits-pilani.ac.in

ticity on the randomness of the output. This paper is organized as follows. The theory of ReRAM and its stochasticity is explained in section 2. The simulation setup is described in section 3. The design and results are discussed in section 4. Analysis of variation in device properties on output is done in section 5. Conclusion from this work are presented in section 6.

2. Theory

2.1. Resistive Switching Random Access Memory (ReRAM)

ReRAM is a two-terminal, non-volatile emerging memory device belonging to the family of memristive devices [20, 21]. A memristor, derived from "Memory" and "Resistor," is a two-terminal device whose resistance equals the total amount of charge flown through it. Consequently, the resistance of a ReRAM can be controlled by applying a voltage across the electrodes, and the device can retain its state until an appropriate voltage is applied to change the state. ReRAM consists of a Metal-Insulator-Metal (MIM) stack where the insulator is generally metal oxide. The device works on the principle of ion migration, where ions migrate through the insulator from one terminal to the other, forming a conductive filament when voltage is applied. ReRAM has two states – Low Resistance State (LRS) and High Resistance State (HRS). The conductive filament, formed by the migration of ions, provides a path for current to flow between the filaments, setting the device in the LRS. Switching from HRS to LRS is known as setting the device, and the voltage at which it occurs is known as set voltage. The device is reset when it switches from LRS to HRS; the applied voltage is known as reset voltage. When the magnitude of the applied voltage is greater than the magnitude of the reset voltage, the conductive filament is ruptured. When the magnitude of the applied voltage is less than the set or reset voltage, the device retains its state. The I-V graph of a typical ReRAM device is shown in Figure 1. The state of the device can be sensed by applying a read voltage less than the set/reset voltage and measuring the current.

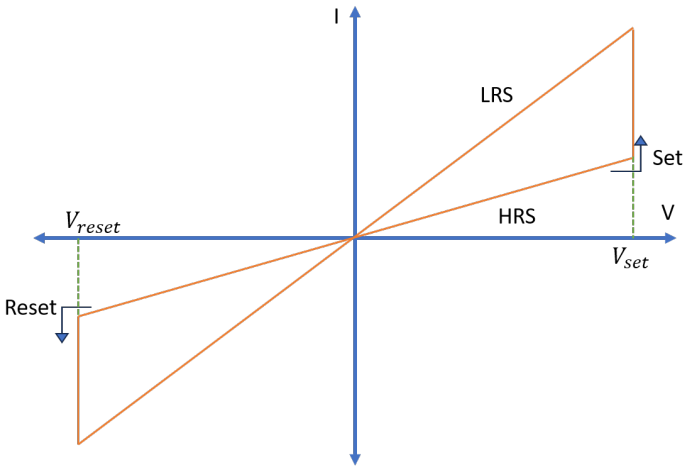


Figure 1: I-V Graph of a typical ReRAM

2.2. Stochasticity in ReRAM

ReRAMs suffer broadly from two types of stochasticity – Dynamic and Static. Dynamic stochasticity is observed during the switching of the states, and variability can be observed in switching voltages and the time required for the device to switch from one state to another [22]. The probability of switching is also random and follows a lognormal distribution [23]. The switching probability increases with an increase in programming amplitude and time for which the voltage pulse is applied. Static stochasticity is the variability in the final resistance value of the device in LRS and HRS. This variability closely follows a lognormal probability density function [24, 25, 26] and hence is modeled as such. The cycle-to-cycle variation in resistance values and switching probabilities is because the filament formation and rupture cannot be precisely controlled in every cycle. The filament's width and length vary from one cycle to another. This is more significant in HRS as the filament length, after breaking, can take up any value as long as it is disconnected from the terminal. This is observed in the device's resistance values, as the resistance variation is much more significant in HRS than in LRS [18]. The inherent dynamic and static stochasticity can be exploited to extract random numbers. The time or voltage required to switch is used in many proposed circuits, but as mentioned earlier, precise control of applied voltage and pulse timing is required, which makes the design complicated. Extracting random bits using static stochasticity is easier because the device is in a stable state, and as long as these states are reached, there is no need for precise control of the input signals. We exploit the significant variance in HRS resistance stochasticity in our proposed design.

3. Simulation

The working of the proposed design was verified by simulation, and further analysis of the variation of device parameters on the randomness of the output was also done. To simulate the ReRAM device, we used the Stanford-PKU RRAM Model [27]. The device is written in Verilog-A and modeled using an internal variable that corresponds to the length of the conductive filament in a device. While a device may have multiple filaments between the two terminals, the model uses a single filament, which acts as a cumulation of all the filaments. The increase in the internal variable corresponds to the growth of filament, and the decrease corresponds to decay. The change in the variable is dependent on the voltage across the terminal. To ensure that the device switches states, the set and reset voltages are set to 2 and -2 volts, respectively, greater than the set and reset voltage of the device, and the read voltage is set to 0.5 volts. The switching behavior of the model is shown in Figure 2.

The resistance of the device is dependent on the gap (g) between the end of the conductive filament and the terminal opposite to the temperature and is given by (1).

$$g = L - l \tag{1}$$

L is the device length, and l, the internal variable, is the length of the filament. If the read voltage is kept constant for the model, the device's resistance is exponentially proportional to g. In other words, the device's resistance in HRS increases exponentially with an increase in g, as shown in Figure 3. A random value of g is

picked from a normal distribution given by (2) to simulate the cycle-to-cycle variation in the device's resistance.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{x-\mu}{\sigma}^2} \tag{2}$$

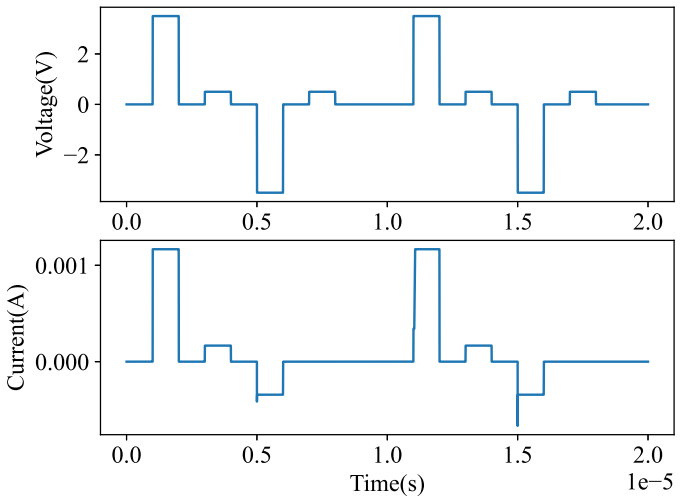


Figure 2: Switching of the states in Stanford-PKU RRAM Model.

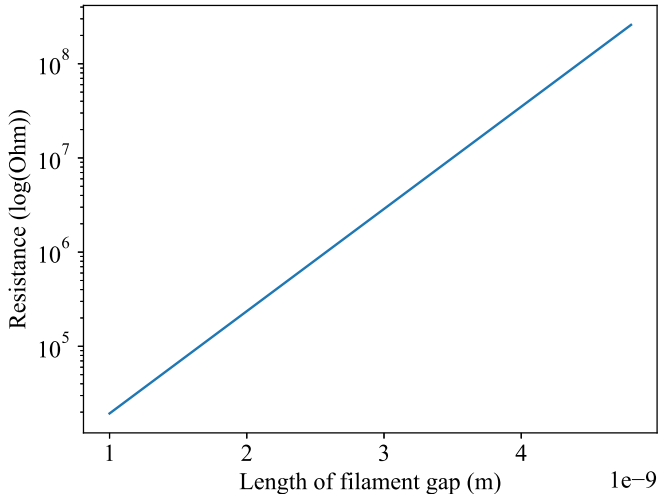


Figure 3: Relation between HRS resistance and g

μ is the mean of the distribution, and σ is the standard deviation. The variation of the random values can be changed by tweaking the values of μ and σ. For the initial simulations, μ was set to 3 nm, and σ was set to 0.1 nm. Since the device's resistance is exponentially related to g, it follows a log-normal distribution when g follows a normal distribution. The cycle-to-cycle variation of HRS for 10000 cycles is shown in Figure 4 and matches the trend followed by the device in [18]. To verify the proposed design, we have picked the same μ and σ for all the devices. The effect of different μ and σ on the output is studied in section 5.

The design requires other circuit components like switches, diodes, and a current direction sensor. We wrote Verilog-A codes for the ideal behavior of these circuits for simulation. The ideal components help us verify the working of our proposed design without affecting the working principle. The switches were modeled after transmission gates controlled by an external voltage source. The diodes have a forward bias voltage drop of 0.7 V. The current direction sensor is programmed to output 1 when the current is positive and 0 when the current is positive.

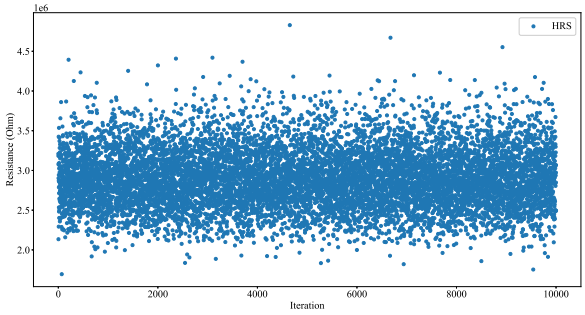


Figure 4: Distribution of HRS and LRS resistance for 10000 set-reset cycles

4. Design and Results

4.1. Working Principle

The working principle for the proposed design is based on the proposed circuit in [18]. In every cycle, two devices are set and then reset to HRS. The devices independently acquire a random resistance value from a log-normal distribution. The resistance values of these two devices are then compared, and the output bit is decided depending on which of the devices has greater resistance. The resistance value in HRS is used because the resistance variation is more significant than LRS.

4.2. Single Bit Design

Our primary aim was to propose a design compatible with a ReRAM crossbar. The proposed design, shown in Figure 5, utilizes a single column of the crossbar and generates one bit per cycle. The design uses two ReRAMs (M1, M2) as the source of randomness and one ReRAM (M3) for bit extraction (explained later). The design uses transmission gates (T1-T5), controlled by voltage sources (C1-C5), as switches. The transmission gates connect the devices to different voltage sources and ground. The design also uses current sensors that sense the current flow direction. The TRNG operation consists of the following steps:

1. One terminal of all three ReRAMs, M1, M2, and M3, is connected to the ground, and the devices are set into LRS by applying a set voltage of 2 V to the other terminal of the devices.

2. All three devices are disconnected from the ground. One of the terminals of M1 and M2 is connected to one of the terminals of M3. The other terminals of M1 and M2 are connected

to their respective voltage sources, and the other terminal of M3 is connected to the current sensor.

3. Read voltage of magnitude 500 mV, and opposite amplitude is applied to M1 and M2 through the voltage sources.

4. The voltage at the common terminal of M1 and M2 is given by the (3), where R1 and R2 is the resistance of M1 and M2 respectively.

$$V = V_{read}\frac{R_2 - R_1}{R_2 + R_1} \tag{3}$$

The voltage is positive and negative depending on the resistance values of M1 and M2, and so is the current direction through the current sensor, given by (4), where R3 is the resistance of M3.

$$I = \frac{1}{R_3}V_{read}\frac{R_2 - R_1}{R_2 + R_1} \tag{4}$$

The current is positive (negative) if the resistance of M1 is smaller (greater) than the resistance of M2.

5. The output bit is decided by the direction of the current sensed by the current sensor. The output bit is 1 if the current is positive and 0 if it is negative.

6. All the ReRAMs are again set to LRS for the next cycle.

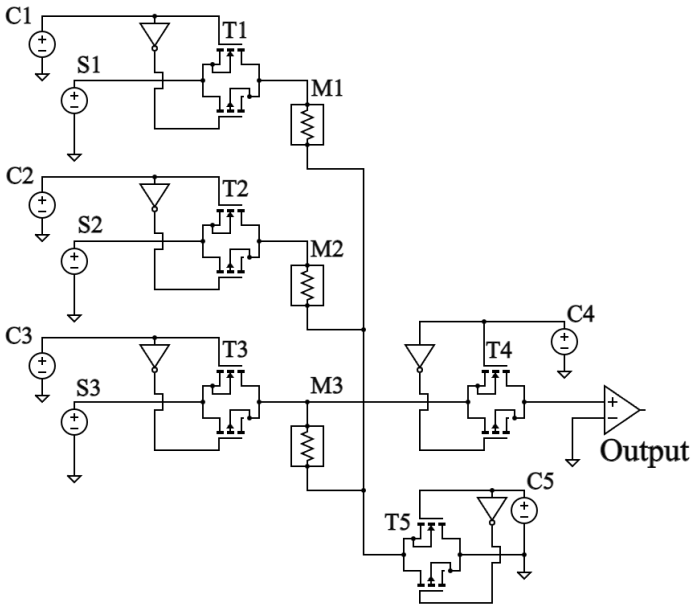


Figure 5: Proposed single-bit design which uses one column of a ReRAM crossbar

The working of the circuit can be seen in Figure 6. The gap, g, and hence the resistance of M1, is lower in cycle one and greater in cycle two than M2. The current through the current sensor is positive and negative in cycles 1 and 2, respectively, as predicted.

4.3. Multi-bit Design

The same principle can be extended to multiple columns in parallel to extract multiple bits in the same cycle. The bits can be read primarily in two ways. Read voltage can be applied multiple times while reading from different columns each time. Or, the bits can be read simultaneously. The second option will consume less time but require more hardware for parallel operation. For verification purposes, we read the output from each column one after the other by applying multiple read signals. The multi-bit design is implemented using a 2x3 ReRAM crossbar and one row of read ReRAMs, considered part of the peripheral circuit, as shown in Figure 7. The design produces three bits per cycle.

The main challenge with using multiple columns is the sneak path current from one column to another, affecting the output bits. We added diodes in the read row to prevent the sneak path current. The diodes prevent the flow in the reverse direction because it is in reverse bias, and since the forward bias voltage is less than the threshold voltage of the diode, no current flows in the forward direction as well. The set voltage applied to the read row is increased to ensure that all ReRAMS are set. The number of bits generated per cycle can be easily increased by increasing the number of columns. However, the number of columns will be limited by the maximum voltage that can be applied as the set voltage for the read row. Also, multiple applications of read voltages in a single cycle may affect the result of the later columns as the devices in these columns may change their state.

4.4. Results and Discussion

Determining the randomness of a sequence of numbers is a challenging task. Generally, a sequence must pass a set of statistical tests to be considered random. We use the NIST SP 800-22 [19] suite of statistical tests to test the sequence generated during the single-bit and multi-bit design simulations. The suite consists of various tests, and a p-value is calculated for each test. If the p-value exceeds 0.01, the sequence passes that particular test. 10,000 bits were generated; their test results are shown in table 1 for single-bit and multi-bit. The generated bit stream passed all the major tests.

The results show that our design can produce a sequence of random numbers. One point to note is the use of ideal switches, diodes, and current sensors for the simulation. We assume that replacing the ideal devices with practical ones will not affect the function of the circuit as long as we ensure that the ReRAMs switch their states, as the design only concerns the final state of the device. The practical devices will mainly affect the set and reset voltages to be applied. This also makes the design immune to variability in threshold voltage and switching time. This flexibility allows the circuit to work with any device as long as the device shows variation in one of the stable states.

The major benefit of the design is that it eliminates the need for additional circuitry to generate random bits. Whenever random bits are required, they can be generated in situ by dedicating some columns of a crossbar for generation. While designing a multi-bit circuit, the designer has the freedom to choose between the number of bits generated per cycle and time per cycle, depending on the constraints.

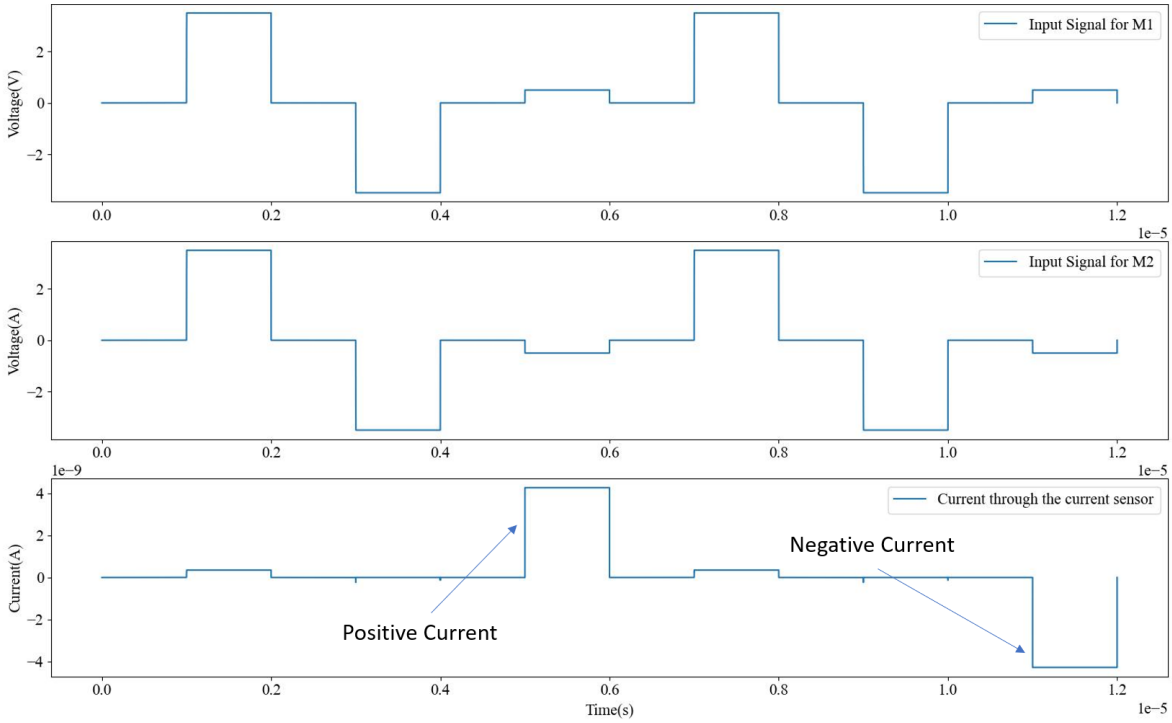


Figure 6: Working of the circuit.

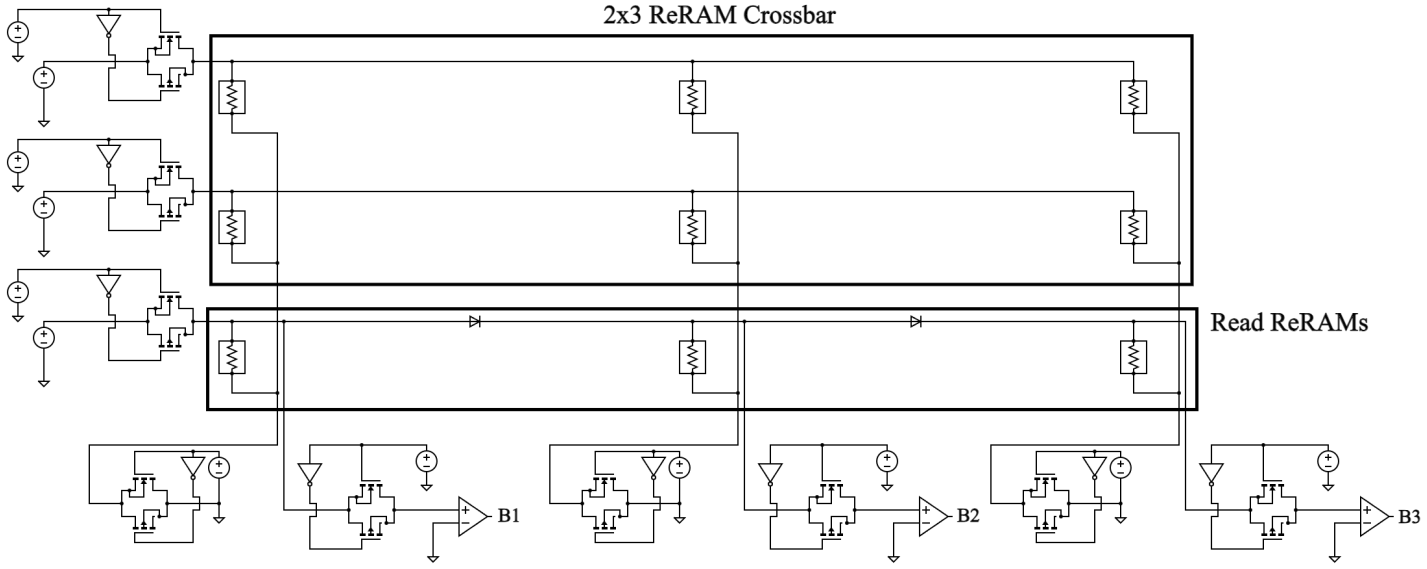


Figure 7: Proposed multi-bit design which uses a 2x3 ReRAM crossbar and a row of read ReRAMs.

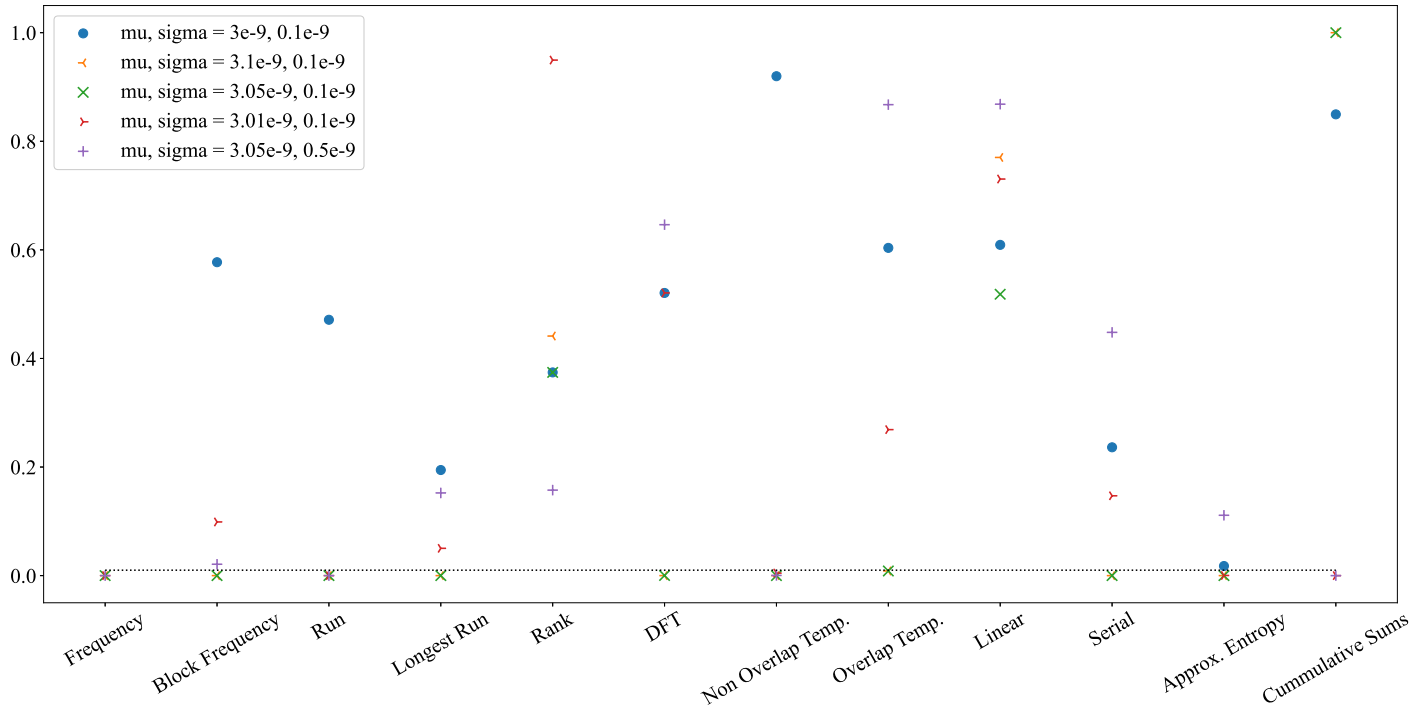


Figure 8: NIST Test Results for different values of μ and σ.

Table 1: NIST Test Result for Single and Multi Bit Circuit

Test	Single Bit		Multi Bit	
	p-value	***Result***	***p-value***	***Result***
Frequency	0.825	Random	0.355	Random
Block Freq.	0.577	Random	0.356	Random
Run	0.471	Random	0.591	Random
Long Run	0.194	Random	0.932	Random
Rank	0.374	Random	0.368	Random
DFT	0.520	Random	0.710	Random
Non-Overlap Temp.	0.919	Random	0.221	Random
Overlap Temp.	0.603	Random	0.932	Random
Linear	0.609	Random	0.147	Random
Serial	0.236	Random	0.368	Random
Approx. Entropy	0.0177	Random	0.586	Random
Cumm. Sum	0.849	Random	0.651	Random

5. Analysis of Statistical Variation

The output's randomness depends on the device properties' stochastic variation. The proposed design involves two devices simultaneously to extract the random bit. The statistical parameters for the random distribution, μ and sigma, were matched for the two devices to verify the working of the circuit. It is also essential to see the effect on the output's randomness if these values are mismatched for the two devices. This analysis is critical to understanding the limitations of the circuit design because of device-to-device variation during fabrication. Bits were extracted by changing the μ and σ of one of the devices, and the randomness of the bit stream was tested using the NIST test suit. The results of different tests are shown in Figure 8.

First, the effect of different mean distances (g) for the two devices was checked by increasing the μ for one device by 3.33%. As seen from the graph, the extracted bits fail to pass most of the tests. Even after decreasing the increase in μ to 1.67%, the bit stream does not pass most tests. Finally, when μ is increased by just 0.33%, the device's output passes most of the test. It can be concluded that the output is very sensitive to device mismatches. The circuit can only tolerate a very low difference in the mean of the gap before it starts generating a non-random output. Thus, very close attention must be paid to device mismatch while fabricating the circuit. An interesting observation is made when the σ of the distribution is also changed when changing μ. Increasing the σ by 400% when the μ of one of the devices is increased by 1.67%, results in the output passing more tests. Hence, a more significant cycle-to-cycle variation can tackle a greater device-to-device variation. While a greater variation is detrimental to most circuits, it benefits the proposed circuit.

6. Conclusion

The proposed TRNG uses inherent randomness in the resistance value of HRS to generate random bits. The design is entirely implementable in ReRAM crossbars. The resistance value of two ReRAMs in HRS in a crossbar is compared, and the output bit depends on their relative values. Circuits for generating both one and multi-bit per cycle are suggested. The circuits were simulated, and the generated bit stream passed almost all NIST randomness test suite tests. The design allows for choosing operating parameters without changing the hardware and will be compatible with various types of ReRAM. Significant device-to-device variability results in

the output bit stream being not random. The effect can be negated by a more significant cycle-to-cycle variation, which is unsuitable for other applications but positively impacts the random number generation application.

Future work will focus on implementing the design on actual hardware and validating the functioning of the design. It will be crucial to study whether the output is affected when the ideal devices are replaced with actual devices and, if so, how. The effect of adjacent columns on the output is also a potential scope of study.

References

[1] T. Patni, A. Pethe, "True Random Number Generator Implemented in ReRAM Crossbar Based on Static Stochasticity of ReRAMs," *2023 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, **7**:55–59, 2023, DOI: 10.1109/APCCAS60141.2023.00024

[2] P. L'Ecuyer, "Random numbers for simulation," *Commun. ACM*, **33**, 10:85–97, 1990, DOI: 10.1145/84537.84555

[3] A. J. Menezes, S. A. Vanstone, P. C. Van Oorschot, *Handbook of Applied Cryptography (1st. ed.)*, CRC Press, Inc., USA, 1996

[4] D. Eastlake, J. Schiller, S. Crocker, "RFC4086: Randomness Requirements for Security," *RFC*, 2005, https://tools.ietf.org/html/rfc4086

[5] Z. Gutterman, B. Pinkas, T. Reinman, "Open to Attack: Vulnerabilities of the Linux Random Number Generator," *Black Hat*, 2006, https://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Gutterman.pdf

[6] J. Kelsey, B. Schneier, D. Wagner, C. Hall, "Cryptanalytic Attacks on Pseudorandom Number Generators," *Fast Software Encryption*, *FSE 1998*, *Lecture Notes in Computer Science*, **1372**:12, Springer, Berlin, Heidelberg, 1998, DOI: 10.1007/3-540-69710-1_12

[7] L. Gong, J. Zhang, H. Liu, L. Sang, Y. Wang, "True Random Number Generators Using Electrical Noise," *IEEE Access*, **7**:125796–125805, 2019, DOI: 10.1109/ACCESS.2019.2939027

[8] A. Vassilev, T. Hall, "The Importance of Entropy to Information Security" *Computer*, **47**, 02:78–81, 2014, DOI: 10.1109/MC.2014.47

[9] Z. Liu, D. Peng, "True random number generator in RFID systems against traceability," *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference*, 620–624, 2006, DOI: 10.1109/CCNC.2006.1593098

[10] F. Pareschi, G. Setti, R. Rovatti, "Implementation and Testing of High-Speed CMOS True Random Number Generators Based on Chaotic Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, **57**, 12:3124–3137, 2010, DOI: 10.1109/TCSI.2010.2052515

[11] M. Park, J. C. Rodgers, D. P. Lathrop, "True random number generation using CMOS Boolean chaotic oscillator," *Microelectronics Journal*, **46**, 12, Part A:1364–1370, 2015, DOI: 10.1016/j.mejo.2015.09.015

[12] N. Nguyen, G. Kaddoum, F. Pareschi, R. Rovatti, G. Setti, "A fully CMOS true random number generator based on hidden attractor hyperchaotic system," *Nonlinear Dyn*, **102**:2887–2904, 2020, DOI: 10.1007/s11071-020-06017-3

[13] F. Zahoor, T. Z. Azni Zulkifli, F. A. Khanday, "Resistive Random Access Memory (RRAM): an Overview of Materials, Switching Mechanism, Performance, Multilevel Cell (mlc) Storage, Modeling, and Applications," *Nanoscale Res Lett*, **15**:90, 2020, DOI: 10.1186/s11671-020-03299-9

[14] F. Zahoor, F. A. Hussin, U. B. Isyaku, S. Gupta, F. A. Khanday, A. Chattopadhyay, H. Abbas, "Resistive random access memory: introduction to device mechanism, materials and application to neuromorphic computing," *Discover Nano*, **18**:36, 2023, DOI: 10.1186/s11671-023-03775-y

[15] H. Jiang, D. Belkin, S. E. Savel'ev, S. Lin, Z. Wang, Y. Li, S. Joshi, R. Midya, C. Li, M. Rao, M. Barnell, Q. Wu, J. J. Yang, Q. Xia, "A novel true random number generator based on a stochastic diffusive memristor," *Nat Commun*, **8**:882, 2017, DOI: 10.1038/s41467-017-00869-x

[16] B. Yang, D. Arumí, S. Manich, Á. Gómez-Pau, R. Rodríguez-Montañés, M. B. González, F. Campabadal, L. Fang, "RRAM Random Number Generator Based on Train of Pulses," *Electronics*, **10**:1831, 2021, DOI: 10.3390/electronics10151831

[17] J. Postel-Pellerin, H. Bazzi, H. Aziza, P. Canet, M. Moreau, V. D. Marca, A. Harb, "True random number generation exploiting SET voltage variability in resistive RAM memory arrays," *2019 19th Non-Volatile Memory Technology Symposium (NVMTS)*, 1-5, 2019, doi: 10.1109/NVMTS47818.2019.9043369

[18] T. Zhang, M. Yin, C. Xu, X. Lu, X. Sun, Y. Yang, R. Huang, "High-speed true random number generation based on paired memristors for security electronics," *Nanotechnology*, **28**:455202, 2017, doi: 10.1088/1361-6528/aa8b3a

[19] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, S. Vo, "SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *Technical Report*, National Institute of Standards & Technology, Gaithersburg, MD, USA, 2010

[20] L. O. Chua, S. M. Kang, "Memristive devices and systems," *Proceedings of the IEEE*, **64**, 2:209-223, 1976, doi: 10.1109/PROC.1976.10092

[21] T. Prodromakis, C. Toumazou, "A review on memristive devices and applications," *2010 17th IEEE International Conference on Electronics, Circuits and Systems*, 934-937, 2010, doi: 10.1109/ICECS.2010.5724666 10.1109/ICECS.2010.5724666

[22] R. Degraeve, A. Fantini, N. Raghavan, L. Goux, S. Clima, B. Govoreanu, A. Belmonte, D. Linten, M. Jurczak, "Causes and consequences of the stochastic aspect of filamentary RRAM," *Microelectronic Engineering*, **147**:171-175, 2015, 10.1016/j.mee.2015.04.025

[23] G. Medeiros-Ribeiro, F. Perner, R. Carter, H. Abdalla, M. D. Pickett, R. S. Williams, "Lognormal switching times for titanium dioxide bipolar memristors: origin and resolution," *Nanotechnology*, **22**, 9:095702, 2011, 10.1088/0957-4484/22/9/095702

[24] Y. Wang, W. Wen, H. Li, M. Hu, "A Novel True Random Number Generator Design Leveraging Emerging Memristor Technology," *Proceedings of the 25th edition on Great Lakes Symposium on VLSI (GLSVLSI '15)*, 271-276, 2015, 10.1145/2742060.2742088

[25] M. Hu, Y. Wang, Q. Qiu, Y. Chen, H. Li, "The stochastic modeling of TiO2 memristor and its usage in neuromorphic system design," *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 831-836, 2014, 10.1109/ASPDAC.2014.6742993

[26] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, H. S. P. Wong, "Stochastic learning in oxide binary synaptic device for neuromorphic computing," *Frontiers in Neuroscience*, **7**, 2013, 10.3389/fnins.2013.00186

[27] H. Li, Z. Jiang, P. Huang, Y. Wu, H.-Y. Chen, B. Gao, X. Y. Liu, J. F. Kang, H.-S. P. Wong, "Variation-aware, reliability-emphasized design and optimization of RRAM using SPICE model," *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 1425-1430, 2015, 10.7873/DATE.2015.0362